Universidad Autónoma Metropolitana
Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licienciatura en Ingeniería en Computación

Servidor HTTPS en un FPGA

Reporte de proyecto que presenta:
Alejandro Cadena Cervantes

para obtener el titulo de:
Ingeniero en Computación

Asesor:
M. en C. Oscar Alvarado Nava

México, D.F. **Abril de 2014**

II

Resumen

En el presente proyecto se implementó un servidor con protocolo seguro de transferencia de hipertexto, HTTPS (siglas en inglés de *Hypertext Transfer Protocol Secure*) en un sistema embebido basado en FPGA, utilizando la tarjeta de desarrollo XUPV2P de Xilinx

La tarjeta de desarrollo XUPV2P cuenta con un FPGA Virtex II Pro el cual cuenta con dos procesadores incrustados PowerPC 405 por lo que es capaz de ejecutar un sistema operativo. Se compiló e instaló el kernel 3.2.0 de Linux, el cuál fue compilado mediante el uso de una *toolchain* generada por el sistema *Buildroot*. De la misma forma se generó un sistema de archivos para el sistema.

El sistema operativo Linux fué compilado para soportar las bibliotecas del API de OpenSSL para así desarrollar y ejecutar un servidor WEB con un canal de comunicación seguro entre cliente y servidor sobre el protocolo de transporte TCP. El sistema es capaz de procesar información que llega cifrada y enviar una respuesta de la misma forma. Al implementar tanto el sistema operativo como el servicio WEB en un solo chip, ofrece gran seguridad, velocidad y estabilidad en la transferencia de información.

Agradecimientos

- A La Universidad Autónoma Metropolitana Unidad Azcapotzalco.
- A la división de Ciencias Básicas e Ingeniería.
- Al Departamento de Electrónica.
- Al M. en C. Oscar Alvarado Nava por su asesoramiento y definir siempre la correcta dirección de este proyecto.
- A Armando de la Cruz Torres, José Enrique Zetina Moya, Julio Blas y Alberto Rodríguez Sánchez por toda su ayuda y colaboración.
- A mi familia y amigos por siempre estar presentes en la realización de este proyecto y por su apoyo siempre incondicional.

Índice general

Índice de figuras

Capítulo 1

Introducción

En la actualidad son ampliamente utilizadas las redes de computadoras en muchos ámbitos, se implementan desde redes de área local, hasta redes de área amplia y todas estas tienen como finalidad compartir información, recursos u ofrecer servicios. Todo esto se da por medio de un proceso de comunicación donde existe una fuente, un transmisor, un sistema de transmisión, un receptor y un destino. Las organizaciones o usuarios que forman parte de estas redes están continuamente manejando e intercambiando información, esto genera la necesidad de protegerla de amenazas que podrían alterarla durante su transmisión, además necesitamos garantizar que los datos transmitidos sean auténticos.

Una red puede estar amenazada básicamente de dos formas distintas: una la llamaremos amenaza pasiva, en la cual un usuario ajeno ataca e intenta obtener información relativa a una comunicación, y la que llamaremos amenaza activa, en la que, además de obtener información, se le hace alguna modificación a los datos transmitidos o se crea una transmisión falsa.

La tecnología esencial en todas las aplicaciones de seguridad en redes es el cifrado de información y de esta existen dos técnicas fundamentales: cifrado simétrico y el cifrado de clave pública. En este proyecto se busca ofrecer esta seguridad mediante la implementación de una capa de seguridad que este presente en un sistema empotrado que pueda realizar tanto el cifrado como el descifrado de información y que permita el intercambio seguro de información entre un cliente y un servidor WEB.

Un servidor HTTP en un sistema embebido es de gran ayuda para crear servicios HTTP entre servidores y clientes ligeros, es decir, aplicaciones que se ejecutan en sistemas de bajos recursos de cómputo. La utilización de un servidor ligero en un sistema embebido puede ser utilizado para muchas funciones de comunicación, como por ejemplo, indexar archivos, generar reportes o mensajes de depuración del sistema presentados en una forma interfaz regular como una aplicación WEB.

1.1. Objetivos

1.1.1. Objetivo general:

Diseñar e implementar un servidor HTTPS1 sobre un FPGA utilizando OpenSSL2.

1.1.2. Objetivos específicos:

Adaptar un sistema operativo Linux en un FPGA. Seleccionar los módulos necesarios para que el sistema Linux sea capaz de ejecutarse en un sistema de recursos limitados. Implementar el servidor sobre el sistema Linux. Seleccionar y agregar los módulos necesarios para poder ejecutar un servidor WEB [1] ligero. Implementar OpenSSL sobre el sistema Linux. Seleccionar y agregar los módulos necesarios para proporcionar funciones de criptografía en el sistema. Comprobar que se realice la comunicación segura entre un cliente y servidor WEB.

1.2. Justificación

En algunos sistemas la comunicación entre el servidor y cliente debe ser confidencial por lo cual es necesario agregar un capa de seguridad en la transmisión y recepción de la información. Los sistemas de seguridad nos permiten desarrollar varios servicios de seguridad como por ejemplo los sistemas de seguridad de clave pública, autenticación de servidores, integridad de mensajes y opcionalmente autenticación de cliente. El sistema de seguridad de clave pública es el que se utiliza para cifrar los datos que vienen del y van al servidor seguro. Con todo esto agregado podemos obtener un sistema que mantenga la seguridad de la información en las transacciones entre un cliente y un servidor que es lo que se busca en este proyecto, ofrecer un sistema de seguridad sobre una tarjeta de desarrollo que permita eficiencia y rapidez en la comunicación.

1.3. Antecedentes

- Implementación de un firewall utilizando un FPGA.

Este proyecto terminal desarrolla un dispositivo en un FPGA que permite la aplicación de listas de acceso y seguridad en redes de computadoras, así como una interfaz gráfica de usuario que permita la modificación de las reglas de operación del mismo. Particularmente implementa una solución que permite la transmisión y recepción de datos entre una computadora con sistema operativo Linux, y una red de computadoras a través de un FPGA. Al igual que en esta propuesta, pretende implementar un dispositivo que proteja la integridad, confiabilidad y disponibilidad de los datos que se transmiten.

[1] dfdfdfdfdf

- Dispositivo de seguridad implementado en la NetFPGA.

Esta es una propuesta de Proyecto Terminal que pretende implementar un dispositivo de seguridad que proporcione control de acceso a la red y que los dispositivos locales cumplan las políticas de seguridad de una red local antes de acceder a la Internet asi como detectar la transmisión de programas maliciosos a través de ella. Para ello utiliza una tarjeta de desarrollo con un FPGA y sobre el cual se implementa un sistema operativo Linux.

- Analysis of the SSL 3.0 Protocol.

En este artículo se ofrece un análisis técnico detallado de la fuerza criptográfica del protocolo SSL 3.0. Una serie de pequeños defectos en el protocolo y varios nuevos paquetes activos sobre SSL se presentan, además de la posibilidad de ser corregidos sin cambio en la estructura básica del protocolo.

- CryptoForge.

Este es un programa de encriptación para seguridad personal y profesional. Permite proteger la privacidad de archivos, carpetas, y mensajes confidenciales encriptándolos (cifrándolos) con hasta cuatro algoritmos de encriptación robustos. Una vez que la información ha sido encriptada, puede ser almacenada en un medio inseguro, o transmitida por una red insegura (como Internet), y aún así permanecer secreta. Luego, la información puede ser desencriptada (descifrada) a su formato original. CryptoForge es un conjunto de programas para encriptar archivos, carpetas, y email, que le añaden a Windows una robusta encriptación.

1.4. Marco teórico

FPGA

Un FPGA (Field-Programmable Gate Array) es un dispositivo semiconductor que puede ser programado después de haber sido fabricado. En lugar de ser restringido por cualquier función de hardware, un FPGA te permite programarle características y funciones, adaptarlo a los nuevos estándares y reconfigurar el hardware para aplicaciones especificas incluso después de que el producto haya sido instalado. Puedes usar un FPGA para implementar cualquier función lógica que una ASIC (Application-Specific Integrated Circuit) pueda llevar a cabo.

SISTEMA EMBEBIDO

Se entiende por sistemas embebidos a una combinación de hardware y software, sumado tal vez a algunas piezas mecánicas o de otro tipo, diseñado para realizar una función específica. Muchas veces un sistema embebido es un componente de un sistema mucho mas grande. Esta combinación de software y hardware puede ser reemplazada en muchos casos por un circuito integrado que realice la misma tarea pero una de las ventajas de los sistemas embebidos es su flexibilidad. Ya que a la hora de realizar alguna modificación resulta mucho mas sencillo modificar una líneas de código al software del sistema embebido que reemplazar todo el circuito integrado.

KERNEL LINUX

El kernel de Linux es el software que realiza las tareas básicas en un sistema de cómputo tales como la administración de memoria para todos los programas en ejecución y la administración del tiempo del procesador que estos mismos usan, además de interactuar con el hardware que conforma un sistema. Por ejemplo, el kernel hace posibles características estándar de Linux como soporte a multitareas y multiusuario. También maneja comunicaciones con dispositivos como memorias flash o tarjetas de sonido. Los usuarios envían peticiones para acceder a estos dispositivos a través del kernel, que después maneja la tarea de nivel inferior de enviar las instrucciones apropiadas al dispositivo .

TOOLCHAIN

Una toolchain es un conjunto de herramientas de desarrollo de software que son ligadas (o encadenadas) por niveles tales como gcc, binutils y glibc. Opcionalmente una toolchain puede contener otras herramientas tales como un depurador o un compilador para un lenguaje de programación especifico. Casi siempre, la toolchain usada para el desarrollo de sistemas embebidos es una toolchain cruzada, o mejor conocida como compilador cruzado. Todos los programas se ejecutan sobre un sistema anfitrión de una arquitectura especifica (por ejemplo x86) pero produce código binario (ejecutable) para ejecutarse sobre una diferente arquitectura. Esto es llamado compilación cruzada y es la manera típica de construir software embebido.

SSL

Uno de los servicios de seguridad más ampliamente utilizados es el de capa de sockets segura (SSL) y el posterior estándar de Internet conocido como capa de transporte segura (TLS), definido este último en el RFC 2246. SSL es un servicio de propósito general implementado como un con- junto de protocolos que hacen uso de TCP. En este nivel, existen dos elecciones de implementación. Para una completa generalidad, SSL (o TLS) podría suministrarse como parte de la familia de protocolos subyacente y, de esta forma, ser transparente a las aplicaciones. Alternativamente, SSL puede integrarse en paquetes específicos. Por ejemplo, los navegadores Netscape y Microsoft Explorer vienen equipados con SSL y la mayoría de los servidores web implementan este protocolo.

Capítulo 2

Hardware y Software

2.1. Introducción

Este proyecto desarrolla un sistema para ser ejecutado sobre la tarjeta de desarrollo XUPV2P, elegida por sus características de desarrollo que permiten crear sistemas programables de propósito especifico. Este hardware es desarrollado por Xilinx. Xilinx, Inc. introdujo el concepto de FPGAs (Field Programmable Gate Arrays) como un dispositivo lógico programable. Desde casi su creación, se ha reconocido el potencial de usar estos dispositivos para construir lo que ahora se conoce como sistemas embebidos. Xilinx es actualmente uno de los líderes en el diseño y fabricación de FPGAs y CPLDs (Complex Programmable Logic Device), así como de circuitos integrados, tarjetas de desarrollo y herramientas de diseño de software principalmente. Esta fue fundada en Silicon Valley en 1984 con sede en San José, California. Durante el desarrollo de este capitulo describiremos la forma de generar la plataforma hardware necesaria para la tarjeta XUPV2P usando el procesador PPC405 y herramientas para el desarrollo de sistemas digitales sobre FPGAs y CPLDs

2.2. Sistemas embebidos y FPGA

Los modernos proyectos de sistemas embebidos desarrollan máquinas de cómputo que se han convertido en parte integral de nuestra sociedad. Estos sistemas están casi en todas partes, desde teléfonos inteligentes o microondas hasta sofisticado equipo médico o formando parte de algún satélite artificial.

Se entiende por sistemas embebidos a una combinación de hardware y software, sumado tal vez a algunas piezas mecánicas o de otro tipo, diseñado para realizar una función específica. Muchas veces un sistema embebido es un componente de un sistema mucho mas grande.

Esta combinación de software y hardware puede ser reemplazada en muchos casos por un circuito integrado que realice la misma tarea pero una de las ventajas de los sistemas embebidos es su flexibilidad. Ya que a la hora de realizar

alguna modificación resulta mucho mas sencillo modificar una líneas de código al software del sistema embebido que reemplazar todo el circuito integrado.

Un componente clave en estos sistemas es el FPGA (Field-Programmable Gate Array). Informalmente un FPGA puede ser imaginado como una placa vacía en la cual cualquier circuito puede ser configurado. Además, la funcionalidad deseada puede ser configurada en esta placa incluso después de que el dispositivo ha sido manufacturado, instalado o, en algunos casos, después de que el producto ha sido vendido al consumidor. Esto hace al FPGA fundamentalmente diferente de otros Circuitos Integrados (CI). En resumen, un FPGA nos ofrece hardware programable para desarrollar sistemas embebidos.

Los actuales FPGA soportan procesadores, buses, controladores de memoria e interfaces de red y otros periféricos más, todo esto en un simple dispositivo. Existe también la posibilidad de agregarle un sistema operativo como Linux, haciendo que el FPGA se comience a parecer más a una computadora de escritorio en términos de funcionalidad y capacidad, sin embargo, sobre un simple CI.

Nosotros usaremos el término Plataforma FPGA para describir un dispositivo que incluye suficientes recursos y funcionalidad para portar un sistema completo sobre un dispositivo simple, jugando entonces, un rol central en el sistema.

La tarjeta de desarrollo que utilizaremos será "Xilinx University Program Virtex-II Pro" (XUPV2P) la cuál es capaz de ejecutar un sistema operativo. Cuenta con un procesador físico PPC405 de IBM y el FPGA XC2VP30 lo cuál proporciona el suficiente rendimiento para la ejecución de nuestro sistema.

2.3. Plataforma de ejecución FPGA

La tarjeta FPGA XUPV2P de Xilinx tiene las siguientes características relevantes :

- Memoria integrada Bloques de SDRAM.
- Multiplicadores de alto rendimiento incorporados.
- Dos núcleos de procesadores IBM PPC405 incrustados.
- Periféricos VGA.
- Puertos serie.
- Tarjeta de red compatible con el estándar IEEE 802.3 (Ethernet).
- Puerto PS/2.
- Codificador de audio.
- Memoria externa con un slot para memoria DDR hasta 512MB
- LEDs, interruptores y pulsadores.

Las características completas de la Tarjeta están disponibles en: http://www.xilinx.com/univ/xupv2p.htm

2.4. Requerimientos

2.4.1. Material para el proyecto

- Memoria DDR de 256 MB.
- Tarjeta CompactFlash de al menos 256MB.
- Tarjeta FPGA XUPV2P Xilinx.
- Cable de Red.
- Tener instalado Debian 6 (squeeze)en el equipo.
- GIT Software de Gestión de Versiones.
- Cable RS232 serial a USB y Software de comunicación serial.
- Xilinx Platform Studio (EDK) 8.2i.
- Xilinx Integrated Software Environment (ISE) 8.2i.

2.4.2. Minicom

Minicom nos permitirá establecer una consola serie remota para acceder a nuestro sistema en la tarjeta desde nuestra máquina de desarrollo conectándonos por medio del puerto serie de la tarjeta al puerto USB de nuestra máquina de desarrollo con el cable RS232. Minicom es un programa de módem basado en texto y emulación de terminal. La forma de instalarlo es:

: ~# apt-get install minicom

Minicom está configurado por defecto para conectarse a un dispositivo **/dev/ttS1** o a **/dev/modem**. En nuestro caso el dispositivo al que deseamos acceder es **/dev/ttyUSB0**, entonces será necesario configurarlo:

: ~# minicom -s

Tomamos las opciones: **Serial port setu**p y cambiamos el **Serial device**
Para configurarlo: CTRL+A y después Z. La opción P permite cambiar la configuración de acceso

2.4.3. GIT

Git es un sistema de control de versiones distribuido, que permite el desarrollo de proyectos de software de manera escalable, distribuida y no lineal. Para instalar:

: ~# apt-get install git-core

2.5. Software Xilinx ISE y EDK

EDK (Embedded Development Kit) es un ambiente de desarrollo integrado para el diseño de sistemas embebidos. Este paquete incluye el XPS (Xilinx Platform Studio) y el SDK (Software Development Kit) así como también toda la documentación que se requiere para el diseño en FPGAs Xilinx ya sea con un procesador PowerPC o MicroBlaze.

ISE (Integrated Software Environment) es una herramienta producida por Xilinx para la síntesis y el análisis de diseños en HDL (Hardware Description Language). Este habilita la compilación de los diseños, examina los diagramas RTL y simula diseños entre algunas otras cosas.

2.5.1. Instalación

2.5.1.1. Requerimientos

Existen algunas pasos que debemos de seguir antes de iniciar con la instalación del software. El sistema operativo que estamos utilizando es Debian 6 (squeeze) y necesitaremos instalar algunas herramientas básicas de desarrollo:

: ~# apt-get install gcc

: ~# apt-get install g++

: ~# apt-get install build-essential

: ~# apt-get install fxload

También necesitaremos una librería conocida como The GNU Standard C++ Library V3:

: ~# apt-get install libstdc++5

Por ultimo instalaremos una biblioteca que es requerida por algunas herramientas del EDK. A menos de que ya se tenga instalada en el sistema instalaremos de la siguiente forma:

: ~# apt-get install libdb4.6++-dev

Terminada la instalación deberemos agregar los "service packs" cargando primero las variables de entorno

: ~# source /opt/Xilinx/settings.sh

: ~# webupdate

Para desmontar algún dispositivo o imagen se ejecuta el siguiente comando:

: ~# umount /mnt

2.5.1.2. Instalación ISE

La instalación la realizamos a partir de imagen ISO de la versión 8.2i de este software que fue proporcionado por la universidad. Primero se debe montar la imagen para tener acceso a los archivos, esto se hace con el siguiente comando:

: ~# mount -o loop <imagenISO-ISE.iso> /mnt

Ahora podemos iniciar con la instalación:

: ~# /mnt/setup

Los pasos de la instalación podemos seguirla en el Apéndice A.

2.5.1.3. Instalación EDK

La instalación del EDK 8.2i es prácticamente igual que del ISE. Procedemos a montar la imagen de la misma forma que lo hicimos anteriormente para el ISE:

: ~# mount -o loop imagenISO-EDK.iso /mnt

Ahora podemos iniciar con la instalación ingresando a:

: ~# /mnt/setup

Igualmente los pasos de la instalación se encuentran en el Apéndice A.
En esta instalación se creará un directorio llamado EDK en /opt.
Terminada la instalación actualizamos cargando primero las variables de entorno :

: ~# source /opt/Xilinx/settings.sh

: ~# source /opt/EDK/settings.sh

: ~# /opt/EDK/bin/lin/webupdate

Para desmontar algún dispositivo o imagen se ejecuta el siguiente comando:

: ~# umount /mnt

2.5.2. Ejecución del ISE y EDK

Antes de pretender lanzar las aplicaciones es necesario crear las correspondientes variables de ambiente. Para ISE ejecutamos:

: ~$ source /opt/Xilinx/settings.sh

Para EDK:

: ~$ source /opt/EDK/settings.sh

Ya creadas las variables de ambiente se lanza la aplicación, para cada una ejecutamos correspondientemente:

: ~$ /opt/EDK/bin/lin/xps

: ~$ /opt/Xilinx/bin/lin/ise

2.6. Diseño del hardware

Para comenzar con el diseño de nuestro hardware necesitamos preparar el directorio del proyecto. Este estará ubicado en el home del usuario, en le directorio proyecto/xupv2p y dentro de este copiaremos dos elementos necesarios: bsp y lib.

bsp se encuentra dentro del Device Tree Generator. Este es una herramienta del EDK que conecta las características del Automatic BSP Generation a la herramienta XPS. El propósito del device tree generator es producir un device tree file (xilinx.dts) que tiene la información para el diseño del hardware en tu proyecto EDK. Esto evita tener el trabajo de tener que crear un device tree file con un editor de texto lo cual puede ser algo tardado.

El siguiente comando nos proporcionará el device-tree desde el repositorio de Xilinx:

: $\sim$/proyecto/xupv2p$ git clone git://git.xilinx.com/device-tree.git

Es importante mencionar que en este proyecto se utilizó Device Tree Generator version 1.3 ya que con otra versión pueden obtenerse resultados diferentes.

lib puede descargarse de la página de diligent http://www.digilentinc.com/Products/Detail.cfm?Prod= (XUP-V2Pro Pack). Este archivo es necesario para que XPS reconozca las características de la tarjeta. Lo obtenemos y descomprimimos:

: ~/proyecto/xupv2p$ wget http://www.digilentinc.com/Data/Products/XUPV2P/EDK XUP-V2ProPack.zip

: ~/proyecto/xupv2p$ unzip EDK-XUP-V2ProPack.zip

Para nuestro proyecto vamos a necesitar el complemento adicional OPB ETHERNETLITE, el cual podemos obtener desde la pagina de Xilinx.

Para obtener el OPB ETHERNETLITE de la pagina de Xilinx se tiene que crear una cuenta en Xilinx primero. Ya registrado se descarga la licencia como se muestra en la siguiente imagen:

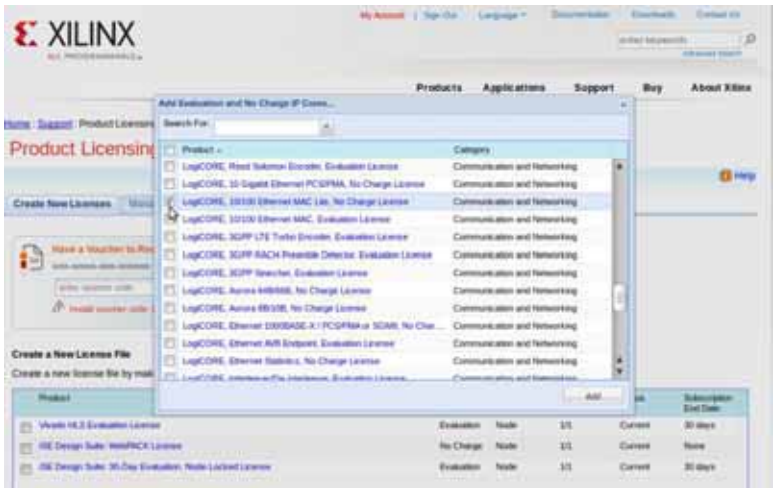

Figura 2.1: Obtener licencia Xilinx

Necesitaremos instalar la licencia copiando el archivo descargado en el EDK de la siguiente manera:

: ~/proyecto/xupv2p$ cp Xilinx.lic /opt/EDK/data/core_licenses

Creación del proyecto

Lo primero que se debe hacer es lanzar la aplicación del EDK, es decir, poner en marcha una nueva sesión de XPS. Habiendo hecho esto elegimos "Base System Builder wizard".

Seleccionamos la ubicación en donde se creará nuestro proyecto (system.xmp). Es importante que esta ubicación sea la misma en donde tenemos nuestro bsp. Además se tiene que hacer uso de un repositorio el cual es el lib que mencionamos anteriormente y contiene las característica de la tarjeta.

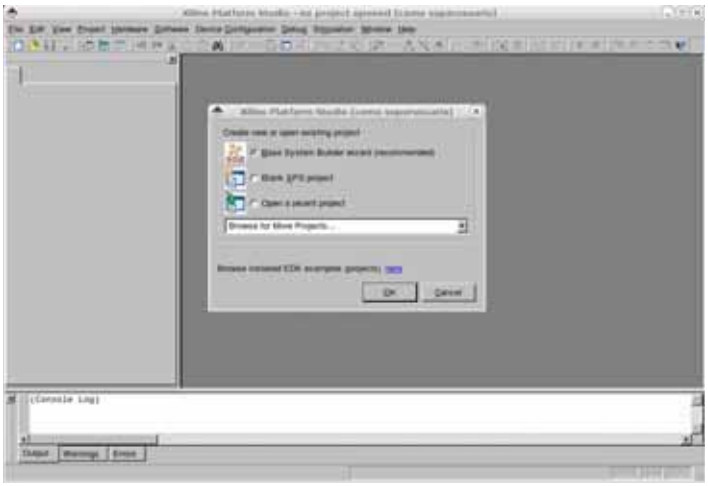

Figura 2.2: Pantalla de Inicio

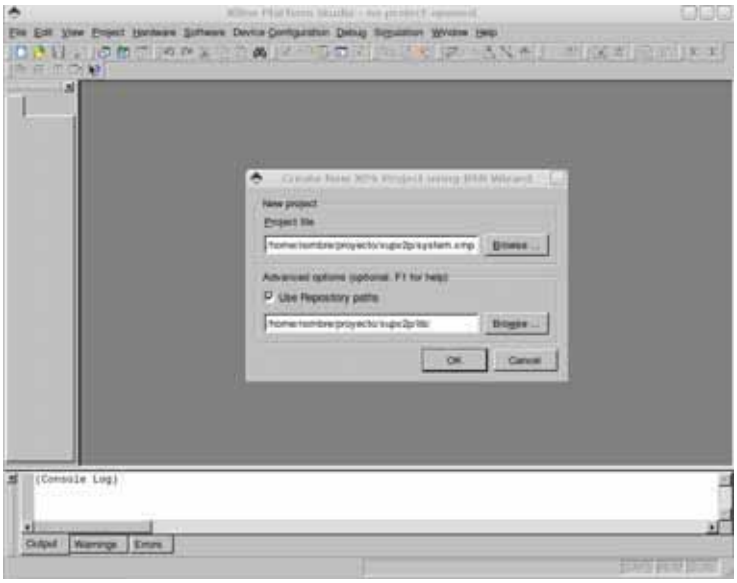

Figura 2.3: Ubicación del proyecto

Habiendo especificado esto ahora le indicaremos al asistente que queremos crear un nuevo diseño seleccionando “I would like to create a new design”.

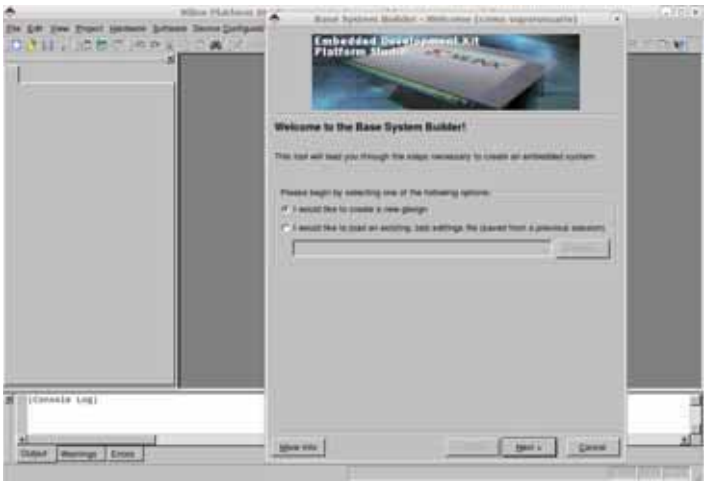

Figura 2.4: Crear nuevo diseño

Ahora debemos seleccionar el tipo de tarjeta que utilizaremos para el desarrollo. Seleccionaremos Xilinx y automáticamente nos aparecerán opciones de nombre y revisión de tarjeta. Seleccionaremos la tarjeta XUP Virtex-II Pro Development System que es la que tiene todas las características que queremos.

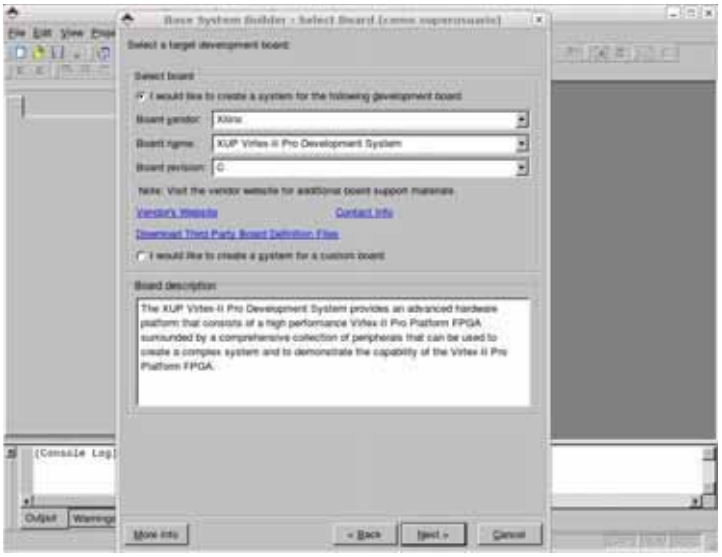

Figura 2.5: Pantalla de selección de tarjeta

Seleccionamos el procesador que usará nuestra tarjeta en este diseño el cual será Power PC.

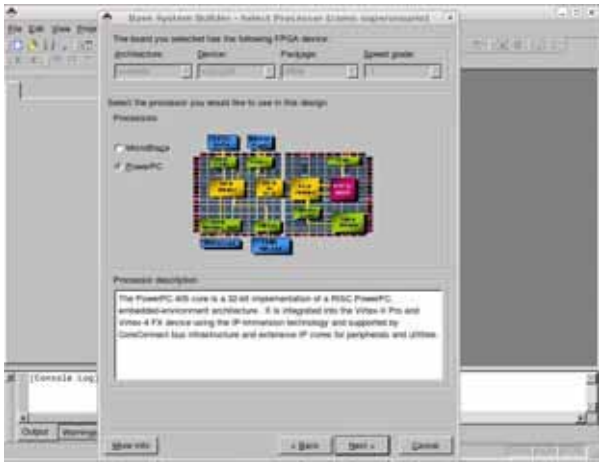

Figura 2.6: Pantalla de selección de procesador

La frecuencia del reloj del procesador sera de 300 MHz y una frecuencia de reloj de bus de 100 MHz. En la configuración del procesador elegiremos JTAG como depurador y habilitaremos la Caché.

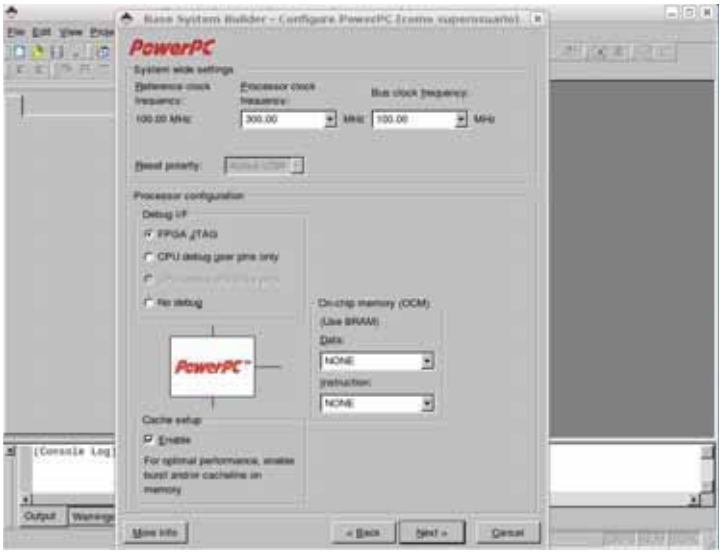

Figura 2.7: Pantalla de selección de características de IBM PPC405

Deshabilitaremos onewire_0 y nos quedaremos con el RS232_Uart_1 con

un Baudrate de 115200. Activamos la interrupción para cada dispositivo que activemos.

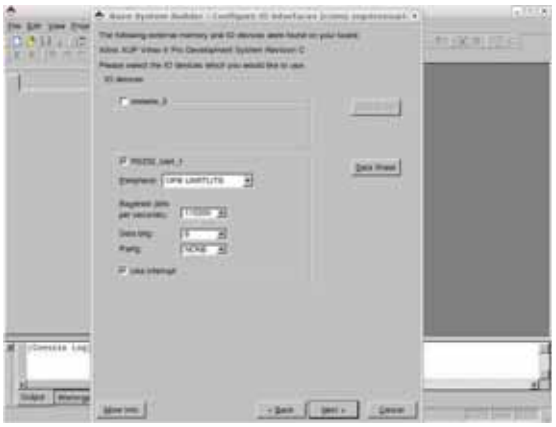

Figura 2.8: Pantalla de configuración de RS232

En la siguiente pantalla, en Ethernet MAC vamos elegir OPB ETHERNETLITE. Dejamos activado SysACE_CompactFlash.

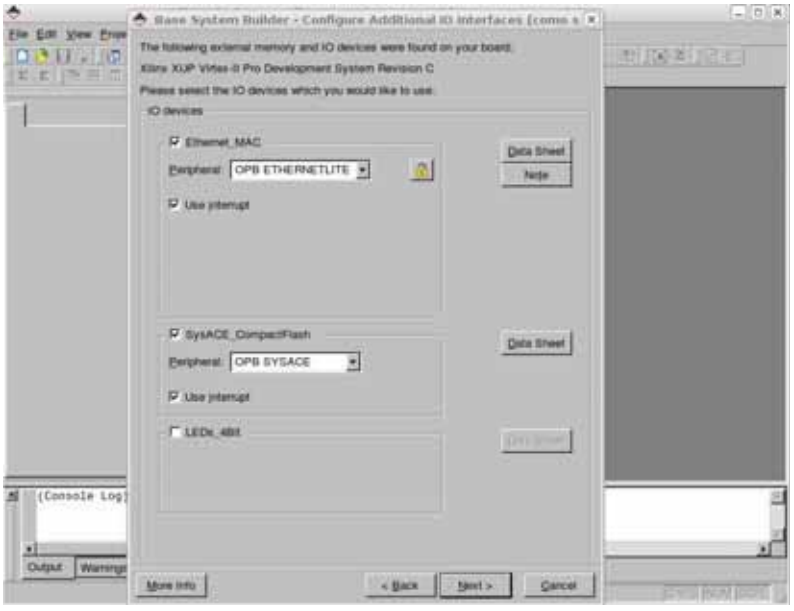

Figura 2.9: Pantalla de selección de MAC Ethernet y SysACE

Deshabilitaremos todos los demás dispositivos ya que no los utilizaremos durante el proyecto.

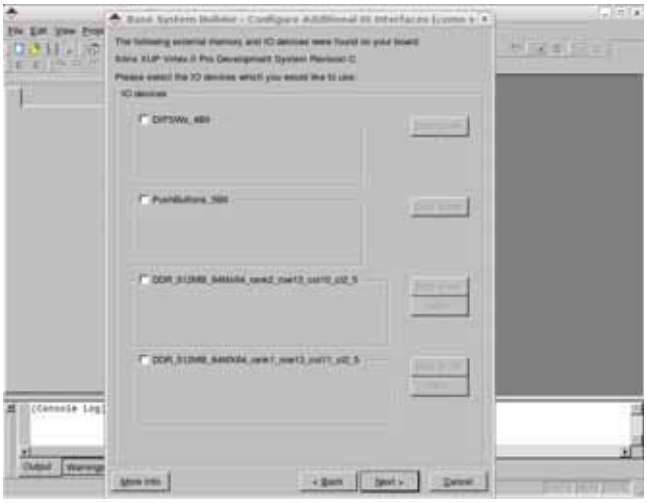

Figura 2.10: Pantalla de selección dedispositivos de memoria DDR

Seleccionaremos la memoria que tenemos disponible. En nuestro caso tenemos disponible 256 MB en RAM. En esta parte no usaremos interrupción. Los demás dispositivos permanecerán deshabilitados.

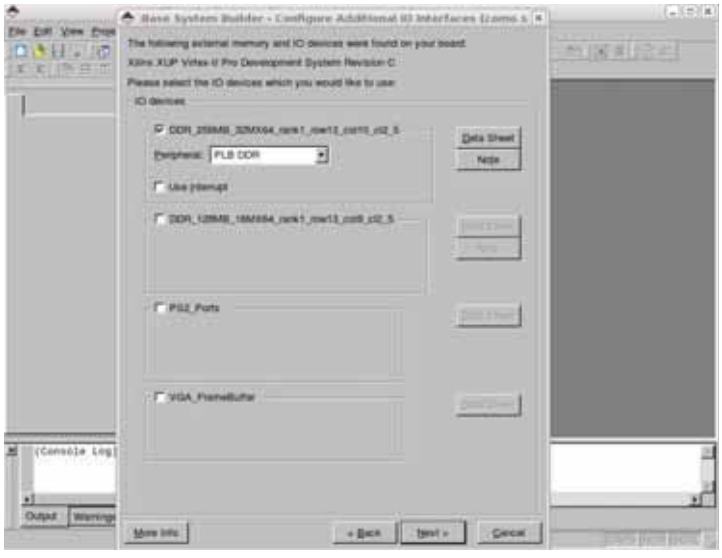

Figura 2.11: Pantalla de selección de dispositivos

Elegimos 128 de BRAM. Esta es necesaria para que el procesador funcione correctamente

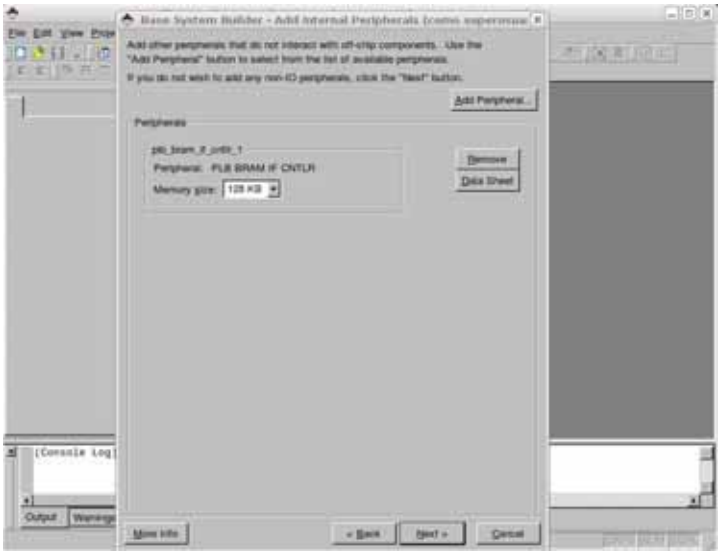

Figura 2.12: Pantalla de configuración de BRAM

Ya que tenemos habilitada la caché nos aparecerá la información de esta, tanto del tamaño de instrucciones como del tamaño de datos. Habilitaremos el Icache y Dcache que son para Instrucciones y datos respectivamente, además de habilitar el Burst. Todo esto solo para DDR.

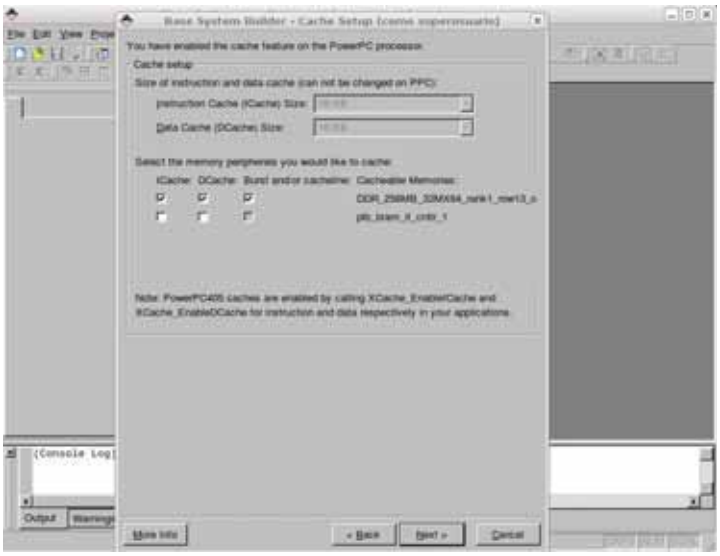

Figura 2.13: Pantalla de selección de líneas de Cache

Habilitamos solo Memory test la cual nos brindará confiabilidad en nuestras memorias.

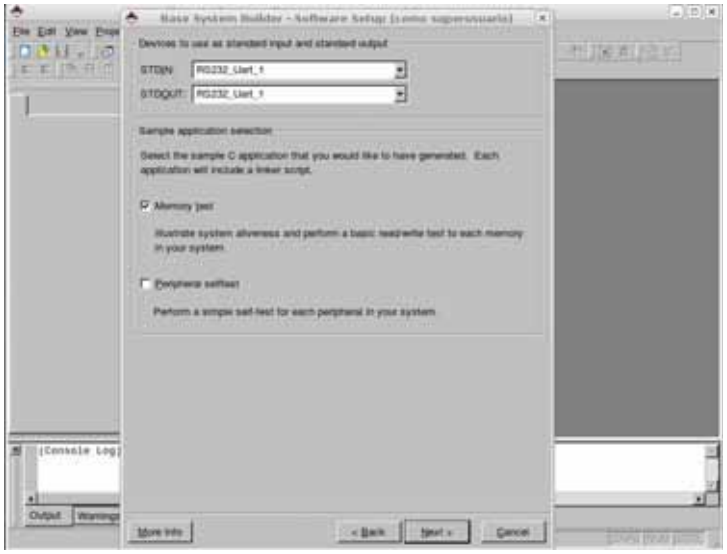

Figura 2.14: Pantalla de selección de progrma de prueba

Dejaremos tanto los datos como instrucciones y Heap en la BRAM.

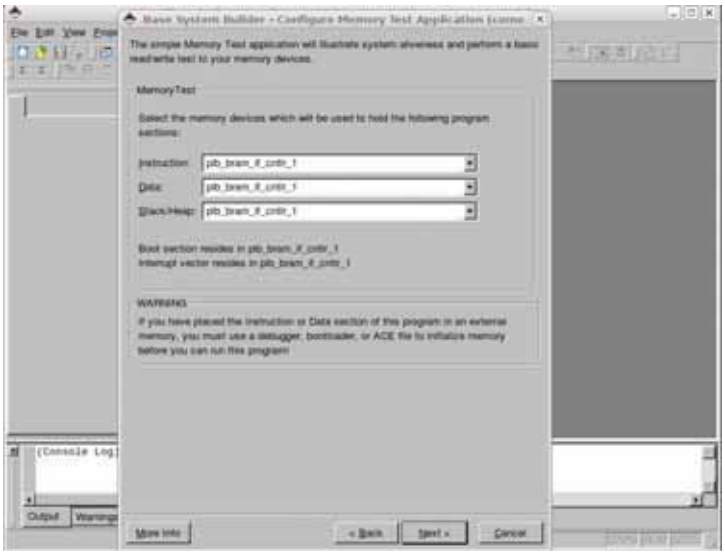

Figura 2.15: Pantalla de selección de división del programa en memoria

Se nos mostrará un resumen del sistema que estamos por crear. Si toda la información es correcta damos clic en Generar.

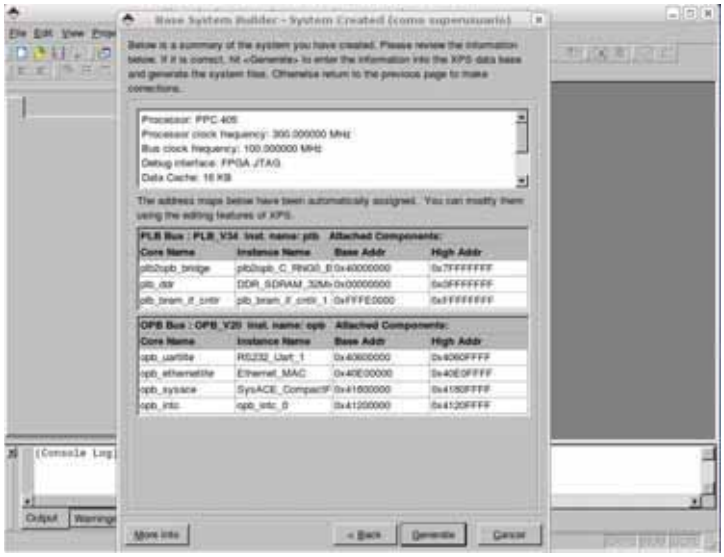

Figura 2.16: Pantalla de generación del hardware

Se nos mostrará el proyecto y ahora debemos hacer unos ajustes. Debemos activar el doble buffer para el núcleo opb_ethernetlite. Para esto se debe dar

doble clic sobre “Ethernet_MAC”.

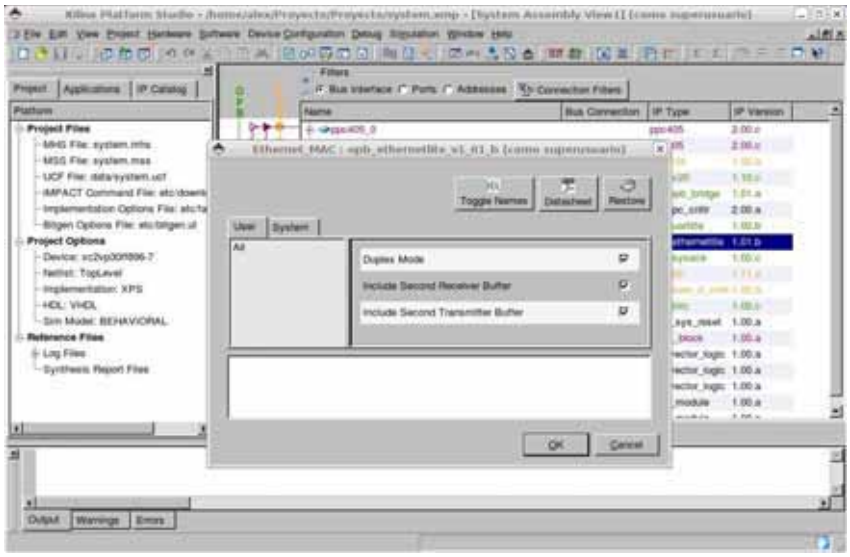

Figura 2.17: Pantalla de configuración de OPB_EthernetLite

Ahora podemos generar el proyecto yendo a Device Configuration -> Update Bitstream.

Con esto también generamos los programas TestApp. Para su ejecución.

Habilitamos el bootloop en la BRAM, dando clic derecho en pestaña de aplicaciones y seleccionando “Mark to initialize BRAM” en los dos casos.

Figura 2.18: Pantalla de configuración de bootloop

Generamos el árbol de dispositivos de Linux. Cuando se haya terminado de ejecutar el software de pruebas básicas , se vuelve a configurar el software en el proyecto de XPS para generar un árbol de dispositivos para compilar un kernel de Linux a la medida.

Hacemos clic en el menú Software -> Software Platform Settings. Seleccionamos “device-tree” como el sistema operativo.

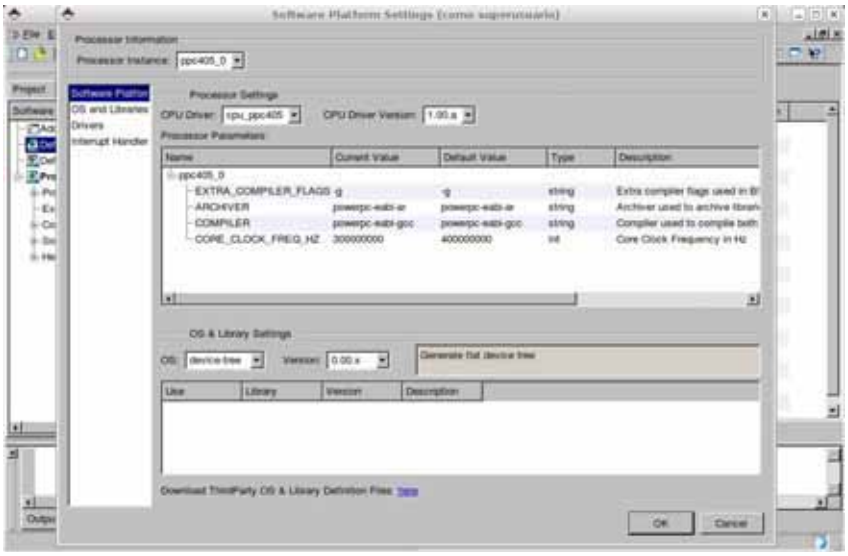

Figura 2.19: selección del device tree

Ahora, en el apartado "OS and Libraries", ajustamos "consola device" a su UART: RS232_Uart_1 y en bootargs sustituimos el valor que esta en Current Value por: "console=ttyUL0,115200 root=/dev/ram rw ip=dhcp".

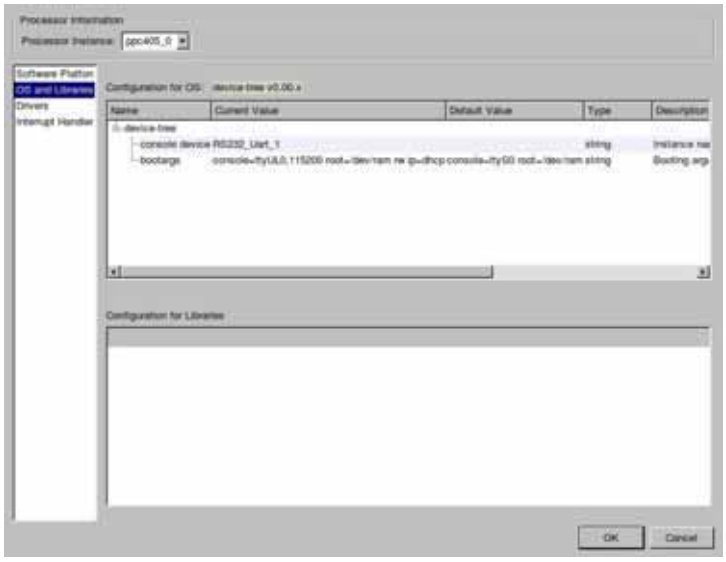

Figura 2.20: Boot Args Kernel Command Line estática

Hacemos clic en "OK". Para generar el árbol de dispositivos hacemos clic en el

menú Software ->Generate Libraries and BSPs. El siguiente archivo se generará dentro del directorio del proyecto EDK: ~/proyecto/xupv2p/ppc405_0/libsrc/device-tree_v0_00_x/xilinx.dts. Este archivo se utilizará para ayudar a construir el kernel de Linux.

El archivo xilinx.dts (Device Tree Hardware Specification) que se genera es similar al siguiente:

```
/*
* Device Tree Generator version: 1.3
*
* (C) Copyright 2007-2008 Xilinx, Inc.
* (C) Copyright 2007-2009 Michal Simek
*
* Michal SIMEK <monstr@monstr.eu>
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU General Public License as
* published by the Free Software Foundation; either version 2 of
* the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston,
* MA 02111-1307 USA
*
* CAUTION: This file is automatically generated by libgen.
* Version: Xilinx EDK 8.2.02 EDK_Im_Sp2.4
*
* XPS project directory: xupv2p
*/
/dts-v1/;
/ {
#address-cells = <1>;
#size-cells = <1>;
compatible = "xlnx,virtex405", "xlnx,virtex";
model = "testing";
DDR_256MB_32MX64_rank1_row13_col10_cl2_5: memory@0 {
device_type = "memory";
reg = < 0x0 0x10000000 >;
} ;
aliases {
ethernet0 = &Ethernet_MAC;
serial0 = &RS232_Uart_1;
} ;
chosen {
bootargs = "console=ttyUL0,115200 root=/dev/ram rw ip=dhcp";
linux,stdout-path = "/plb@0/opb@40000000/serial@40600000";
} ;
cpus {
#address-cells = <1>;
#cpus = <0x1>;
#size-cells = <0>;
ppc405_0: cpu@0 {
clock-frequency = <300000000>;
compatible = "PowerPC,405", "ibm,ppc405";
d-cache-line-size = <0x20>;
d-cache-size = <0x4000>;
dcr-access-method = "native";
dcr-controller ;
device_type = "cpu";
i-cache-line-size = <0x20>;
i-cache-size = <0x4000>;
model = "PowerPC,405";
reg = <0>;
timebase-frequency = <300000000>;
xlnx,dcr-resync = <0x0>;
xlnx,deterministic-mult = <0x0>;
xlnx,disable-operand-forwarding = <0x1>;
xlnx,mmu-enable = <0x1>;
} ;
} ;
plb: plb@0 {
#address-cells = <1>;
#size-cells = <1>;
compatible = "xlnx,plb-v34-1.02.a", "simple-bus";
ranges ;
opb: opb@40000000 {
#address-cells = <1>;
#size-cells = <1>;
compatible = "xlnx,opb-v20-1.10.c", "simple-bus";
ranges = < 0x40000000 0x40000000 0x40000000 >;
Ethernet_MAC: ethernet@40e00000 {
```

```
compatible = "xlnx,opb-ethernetlite-1.01.b";
device_type = "network";
interrupt-parent = <&opb_intc_0>;
interrupts = < 1 0 >;
local-mac-address = [ 00 0a 35 31 dd 00 ];
reg = < 0x40e00000 0x10000 >;
xlnx,duplex = <0x1>;
xlnx,family = "virtex2p";
xlnx,rx-ping-pong = <0x1>;
xlnx,tx-ping-pong = <0x1>;
} ;
RS232_Uart_1: serial@40600000 {
clock-frequency = "";
compatible = "xlnx,opb-uartlite-1.00.b";
current-speed = <115200>;
device_type = "serial";
interrupt-parent = <&opb_intc_0>;
interrupts = < 2 0 >;
port-number = <0>;
reg = < 0x40600000 0x10000 >;
xlnx,baudrate = <0x1c200>;
xlnx,clk-freq = <0x5f5e100>;
xlnx,data-bits = <0x8>;
xlnx,odd-parity = <0x0>;
xlnx,use-parity = <0x0>;
} ;
SysACE_CompactFlash: sysace@41800000 {
compatible = "xlnx,opb-sysace-1.00.c";
interrupt-parent = <&opb_intc_0>;
interrupts = < 0 2 >;
reg = < 0x41800000 0x10000 >;
xlnx,mem-width = <0x10>;
} ;
opb_intc_0: interrupt-controller@41200000 {
#interrupt-cells = <0x2>;
compatible = "xlnx,opb-intc-1.00.c", "xlnx,xps-intc-1.00.a";
interrupt-controller ;
reg = < 0x41200000 0x10000 >;
xlnx,kind-of-intr = <0x6>;
xlnx,num-intr-inputs = <0x3>;
} ;
} ;
} ;
} ;
```

2.7. System.ace

El archivo system.ace permite configurar automáticamente Hardware y software en las tarjetas compatibles con SysACE de Xilinx,para su correcto funcionamiento este archivo se guarda en la primera primera partición de una tarjeta compact flash con formato fat16, el archivo se genera a partir de especificaciones en un archivo de texto plano similar al siguiente:

-jprog
-board user
-target ppc_hw
-hw implementation/download.bit
-elf executable.elf
-configdevice devicenr 1 idcode 0x1127e093 irlength 14 partname xc2vp30
-debugdevice devicenr 1 cpunr 1
-ace system.ace

Donde se ajustan las líneas -hw y -elf para cada situación, el comando para generar el archivo system.ace es el siguiente:

: ∼$ xmd -tcl genace.tcl -opt xupGenace.opt

Este archivo es muy útil para la depuración constante del software sobre una plataforma hardware estable.

Capítulo 3

Compilación cruzada

3.1. Introducción

El kernel es el software que realiza las tareas básicas en un sistema de cómputo tales como la administración de memoria para todos los programas en ejecución y la administración del tiempo del procesador que estos mismos usan, además de interactuar con el hardware que conforma un sistema. Por ejemplo, el kernel hace posibles características estándar de Linux como soporte a multitareas y multiusuario. También maneja comunicaciones con dispositivos como memorias flash o tarjetas de sonido. Los usuarios envían peticiones para acceder a estos dispositivos a través del kernel, que después maneja la tarea de nivel inferior de enviar las instrucciones apropiadas al dispositivo .

Existen herramientas que están a disposición de todos que permiten realizar una configuración del kernel. Esto es para adecuar el kernel a alguna tarea en especifico o simplemente para obtener un mejor rendimiento al momento que este tenga que interactuar con algún hardware en específico. Esto es indispensable cuando hablamos de sistemas embebidos ya que estos son diseñados precisamente para maximizar el rendimiento en dispositivos enfocados a una o varias tareas especificas.

Explicaremos en este capitulo como es que se lleva a cabo una configuración del kernel de Linux y su posterior compilación cruzada para obtener como resultado un sistema embebido a la medida de los requerimientos del objetivo de este proyecto que es la implementación de un servidor https sobre una tarjeta de desarrollo Virtex xupv2p.

3.2. Preparando el kernel

Existen diferente fuentes de donde podemos obtener un kernel de Linux, siendo el sitio oficial http://www.kernel.org/. Para este proyecto utilizaremos el kernel que esta mas apegado a la configuración que necesitamos para trabajar

con la tarjeta xupv2p de Xilinx que es la que estamos utilizando. Este kernel lo obtenemos de la siguiente forma:

: ∼/proyecto$ git clone git://git.xilinx.com/linux-xlnx.git

Ya teniendo nuestro kernel(1) tendremos que copiar dentro de este, en el directorio $\sim$/proyecto/linux-xlnx/arch/powerpc/boot/dts/ el archivo que generamos en nuestro proyecto EDK (xilinx.dts) y que contiene toda la información de configuración necesaria sobre la tarjeta para nuestro proyecto. Este archivo lo renombraremos para que tenga el prefijo "virtex405-":

$cp xilinx.dts ∼/proyecto/linux-xlnx/arch/powerpc/boot/dts/virtex405-nombre.dts

3.2.1. initrd

Para que nuestro kernel funcione correctamente es necesario que tenga un file system el cual lo podemos conseguir desde la página de Xilinx la cual nos proporciona la imagen de un sistema de archivos que son compatibles con el procesador que tenemos en la tarjeta, es decir, con PPC-405:

: ∼/proyecto$ wget http://xilinx.wdfiles.com/local–files/powerpc-linux/ramdisk.image.gz

Colocamos este file system en el directorio boot:

: ∼/proyecto$ cp ramdisk.image.gz $\sim$/proyecto/linux-xlnx/arch/powerpc/boot

Más adelante, en este mismo capitulo, describiremos como es que funciona este sistema de archivos.

3.2.2. Espacio en memoria

Algunas veces, dependiendo del tamaño final del objetivo simpleImage.elf, no se podrá llevar acabo la descompresión total del kernel por espacio insuficiente en memoria, este error se resuelve con lo siguiente:

Abrimos el archivo wrapper:

: ∼/proyecto$ gedit linux-xlnx/arch/powerpc/boot/wrapper &

Cambiaremos el valor de link_address en la linea 146 . Cambiamos de '0x400000' a '0x800000' guardamos y cerramos gedit.

Ahora, ya que contamos con lo fundamental, haremos uso de estos y otros elementos para llevar a cabo la compilación cruzada necesaria para ejecutar el kernel de Linux sobre la FPGA.

3.3. Buildroot

Buildroot es un conjunto de Makefiles y parches que facilitan la generación de sistemas embebidos para Linux. Buildroot puede generar una toolchain, sistemas de archivos, imagenes de kernel y bootloaders. Buildroot es útil principalmente para les personas que trabajan con sistemas con bajos recursos o sistemas embebidos, facilitándoles el proceso de compilación cruzada.

- Se decidió utilizar Buildroot en este proyecto por las siguientes características:
- Puede generar los elementos necesarios para nuestro proyecto de desarrollo de sistema embebido: toolchain, sistemas de archivos, imágenes de kernel y bootloaders
- Es sencillo de utilizar, gracias a que puede configurarse con la interfaz menuconfig.
- Soporta múltiples tipos de sistemas de archivos
- Puede generar una toolchain ya sea del tipo glibc o Clibc.

3.3.1. Toolchain

Una toolchain es un conjunto de herramientas de desarrollo de software que son ligadas (o encadenadas) por niveles tales como gcc, binutils y glibc. Opcionalmente una toolchain puede contener otras herramientas tales como un depurador o un compilador para un lenguaje de programación especifico. Casi siempre, la toolchain usada para el desarrollo de sistemas embebidos es una toolchain cruzada, o mejor conocida como compilador cruzado. Todos los programas se ejecutan sobre un sistema anfitrión de una arquitectura especifica (por ejemplo x86) pero produce código binario (ejecutable) para ejecutarse sobre una diferente arquitectura. Esto es llamado compilación cruzada y es la manera típica de construir software embebido.

Los elementos mínimos para una toolchain son:

- Un cargador (linker)
- Un ensamblador
- Compilador de C
- Bibliotecas de C

Cuando hablamos de toolchain, podemos distinguir tres diferentes máquinas:

- La máquina de desarrollo, sobre la cuál la toolchain es construida.
- La máquina anfitrión, sobre la cuál la toolchain es ejecutada.
- La máquina objetivo, para la cuál la toolchain genera el código.

Desde esas tres diferentes máquinas , nosotros distinguimos cuatro diferentes tipos de procesos de construcción de toolchain:

- *Toolchain nativa*, como la que puede ser encontrada en la mayoría de distribuciones de Linux, normalmente compilada en x86, ejecutada en x86 y que genera código para x86.
- *Toolchian de compilación cruzada*, la cuál es la toolchain más utilizada para el desarrollo de sistemas embebidos. Es normalmente compilada en x86, ejecutada en x86 y genera código para una arquitectura objetivo como puede ser ARM, MIPS, PowerPC, etc.
- *Toolchain cross-native*, Es una toolchain que ha sido construida en x86 pero se ejecuta sobre su arquitectura objetivo y genera código para tu arquitectura objetivo. Es usada cuando quieres un gcc nativo en tu plataforma objetivo sin construirlo en tu plataforma objetivo.
- *Canadian cross*, es el proceso de construir una toolchain en una máquina A para que se ejecute en una máquina B y genere código para una máquina C. Usualmente este tipo no es necesario.

3.3.2. Sistema de archivos

Unix y sus variantes (Linux, BSD, Solaris, etc.) Comparten el concepto de root filesystem (RFS) o sistema de archivos principal. Un sistema de archivos es una estructura de datos implementada como una colección de bloques de memoria con la capacidad de crear, leer y escribir archivos de tamaño variable. Muchos sistemas ofrecen la habilidad de organizar los archivos jerárquicamente.

En Linux, archivos y subdirectorios son agrupados en directorios. Hay un directorio especial llamado root que contiene archivos y subdirectorios; los subdirectorios pueden contener archivos y otros subdirectorios. La estructura de datos del sistema de archivos es implementado en dispositivos de almacenamiento secundario tales como discos duros o discos de estado sólido. Sin embargo, los bloques de memoria pueden ser copiados secuencialmente a otros tipos de memoria, incluyendo RAM, ROM o incluso otro sistema de archivos. Cuando el sistema de archivos esta siendo manipulado de esta forma es llamado comúnmente filesystem image. Cuando el sistema de archivos esta siendo usado para manipular archivos es llamado mounted filesystem.

Para nuestro sistema embebido se debe crear un sistema de archivos inicial llamado root filesystem. A diferencia de algunos sistemas operativos que colocan todo su código de inicio en solo un ejecutable, el proceso de arranque para Linux hace interactuar al kernel con el sistema de archivos desde un inicio. En algunos casos el kernel es almacenado en el sistema de archivos junto con el bootloader. Después el kernel es ejecutado y buscará entre los directorios los archivos de inicio, archivos de configuración del sistema y una aplicación especial llamada init la cual es el primer proceso de arranque del sistema. Este proceso usa los archivos de configuración almacenados en el root filesystem para iniciar los

procesos que le secundan y terminar iniciando desde los archivos de configuración del sistema.

Existen dos formas de crear un filesystem image. En ambos casos se crea un subdirectorio que se convertirá en el root filesystem en el sistema embebido. Esto incluye los archivos de configuración tales como módulos de kernel y librerías. La primer forma es crear un sistema de archivos en una separada partición del disco lo que te permite tratar un archivo como parte de una partición. Una vez que el sistema de archivos es creado, entonces puedes montar el sistema de archivos simplemente copiando tu root filesystem a la nueva locación de montaje. El único inconveniente de este método es que muchos de los pasos requieren privilegios de superusuario. El método alternativo no requiere de root; en lugar de eso usa una aplicación de propósito especifico para generar un filesystem image directamente. Ejemplos de esta aplicación son genisoimage y genext2fs.

En ambos casos, el filesystem image puede ser directamente escrito en algún dispositivo (una partición sobre un sistema embebido, una EEPROM, un dispositivo flash) o combinarlo con el sistema operativo y cargarlo en RAM cuando arranque el sistema. Cuando se copia la filesystem image a RAM y se usa como el root filesystem, la imagen se conoce como "RAM Disk" y ramdisk.image.gz es un nombre de archivo común para una imagen de root filesystem comprimida. Esto permite tener un kernel simple que primero inicia con el RAM Disk, entonces prueba el hardware e instala los módulos del kernel requeridos y finalmente monta el "real" root filesystem. Esto se conoce comúnmente como "initial ramdisk" o "initrd".

En nuestro proyecto utilizaremos primero RAM Disk para probar su funcionamiento y después se montará el sistema de archivos en una partición creada sobre una memoria Compact Flash.

3.3.3. Configuración y compilación

Lo primero será obtener buildroot. Esto se puede hacer directamente en la dirección http://buildroot.uclibc.org/ o con el siguiente procedimiento:

: ~/proyecto$ wget http://git.buildroot.net/buildroot/snapshot/buildroot-2013.02.tar.bz2

: ~/proyecto$ tar -xjvf buildroot-2013.02.tar.bz2

: ~/proyecto$ cd buildroot-2013.02/

Ya que lo tenemos procedemos a configurarlo utilizando menuconfig:

: ~/proyecto/buildroot-2013.02$ make menuconfig

La configuración de buildroot dependerá siempre de las necesidades del diseñador del sistema embebido. Se debe ponerse especial atención en cada uno de los apartados para seleccionar las características que tendrá nuestro sistema

resultante. Tal vez se tenga que realizar varias configuraciones antes de tener el sistema lo que llevará bastante tiempo.

A continuación presentó la configuración básica que se hizo sobre buildroot utilizando menuconfig para este proyecto:

Ejemplo de configuración de buildroot por medio de menuconfig.

```
Target Architecture (PowerPC) —>
Target Architecture Variant (405) —>
Build options —>
($(HOME)/sources) Download dir
Toolchain —>
Toolchain type (External toolchain) —>
Toolchain (Sourcery CodeBench PowerPC 2010.09) —>
Toolchain origin (Toolchain to be downloaded and installed) —
System configuration —>
(xupv2p-nombre) System hostname
(Welcome to nombre) System banner
(password) Root password
(ttyUL0) Port to run a getty (login prompt) on
[ ] remount root filesystem read-write during boot
Package Selection for the target —>
BusyBox Version (BusyBox 1.20.x) —>
(package/busybox/busybox-1.20.x.config) BusyBox configuration file to us
[*] Show packages that are also provided by busybox
[*] Install the watchdog daemon startup script
Compressors and decompressors —>
[*] bzip2
[*] gzip
Debugging, profiling and benchmark —>
[*] stress
Development tools —>
[*] binutils
[*] binutils binaries
[*] tar
Interpreter languages and scripting —>
[*] perl
[*] python
core python modules —>
[*] bzip2 module
[*] zlib module
Libraries —>
Compression and decompression —>
[*] zlib
Crypto —>
-*- ocf-linux
[*] openssl
[*] openssl binary
[*] openssl additional engines
[*] openssl ocf support
Networking —>
[*] glib-networking **
[*] libmnl
[*] libnetfilter_conntrack
[*] libnetfilter_cttimeout
[*] libpcap
Other —>
[*] libcap
[*] libglib2
Text and terminal handling —>
[*] ncurses
[*] readline
Networking applications —>
[*] bridge-utils
[*] dhcpdump
[*] ethtool
[*] iproute2
[*] iptables
[*] mutt
[*] netcat
[*] netstat-nat
[*] openssh
[*] portmap
[*] squid
[*] wget
Shell and utilities —>
[*] bash
Text editors and viewers —>
[*] nano
[*] optimize for size (NEW)
[*] vim
[*] install runtime
Filesystem images —>
[*] ext2 root filesystem
Compression method (gzip) —>
[*] tar the root filesystem (NEW)
Compression method (gzip) —>
```

Al finalizar, la configuración se almacena en un archivo llamado ".config" en este mismo directorio. El archivo de configuración completo se presenta en el apéndice B.

Ya terminada la configuración podemos realizar la compilación:

: ~/proyecto/buildroot-2013.02$ make

La compilación tardará varios minutos dependiendo de el poder de procesamiento de tu máquina. Esta compilación nos generará tanto la toolchain como el sistema de archivos, ambos necesarios para continuar con el desarrollo de este proyecto.

Una vez terminada la compilación podremos hacer uso de la toolchain generada. Para mayor facilidad la moveremos al directorio base de nuestro proyecto:

: ~/proyecto/buildroot-2013.02$ cd output/host/opt

: ~/proyecto/buildroot-2013.02$ mv ext-toolchain/ $\sim$/proyecto

3.4. Compilación del kernel

Entendamos por compilación cruzada al procedimiento mediante el cual podemos generar software para una arquitectura especifica diferente de la que esta llevando a cabo el proceso de compilación. Esto es fundamental para nuestro proyecto y será la manera de generar el sistema que se implementara en nuestra tarjeta de desarrollo.

En primer lugar, es necesario exportar las variables de ambiente para poder usar la toolchain durante la compilación cruzada y para esto tenemos que ingresar las siguientes instrucciones:

unset CC CXX CPP CFLAGS INCLUDES CXXFLAGS LD_LIBRARY_PATH LIBRARY_PATH CPATH

export CC CXX CPP CFLAGS INCLUDES CXXFLAGS LD_LIBRARY_PATH LIBRARY_PATH CPATH

unset ARCH

export ARCH=powerpc

unset CROSS_COMPILE

export CROSS_COMPILE=powerpc-linux-gnu-

export PATH=/ruta /hacia/proyecto/toolchain/bin:$PATH

Estas variables de ambiente especifican que realizaremos una compilación cruzada con una arquitectura objetivo PowerPC, la toolchain que usaremos y la ruta de sus binarios . Las podemos agregar a un archivo (que en este caso se llamara ambiente.sh) y ejecutarlo con el comando source.

En la consola tecleamos:

: ~/proyecto$ source toolchain/ambiente.sh

Volvemos a cargar las variables de ambiente correspondientes a **ISE** y **EDK**:

: ∼/proyecto$ source /opt/Xilinx/settings.sh

: ∼/proyecto$ source /opt/EDK/settings.sh

3.4.1. Kernel Command Line

Durante el proceso de arranque del kernel, el programa principal del kernel inicializa los elementos de bajo nivel, después se observan los parámetros que se pueden pasar de manera estática o dinámica en la llamada "Kernel Command Line"(KCL), estos parámetros son cadenas donde se especifican los valores (binarios o específicos) para distintos elementos del kernel, como por ejemplo cual sera el dispositivo de consola (Donde se encontrara la entrada y salida estándar), donde esta el root filesystem o el programa init, entre muchas opciones .

Un ejemplo de Kernel Command line es el que sigue:

console=ttyUL0,115200 ip=dhcp root=/dev/nfs

Este KCL dice que el dispositivo de consola sera el Serial UartLite (RS232) a 115200 baudios, obtendrá una dirección de red por DHCP y el root filesystem se montara por NFS (Network File System). Esta KCL se especifica en "Kernel Options"de la configuración del kernel o en el árbol de dispositivos (Device Tree) teniendo prioridad este ultimo.

3.4.2. Configuración

Una vez establecidas las variables de ambiente se procede a configurar el kernel de Linux:

: ∼/proyecto$ cd linux-xlnx

: ∼/proyecto/linux-xlnx$ make menuconfig

Con esto se nos desplegará la interfaz de menuconfig donde estableceremos la configuración del kernel. Esta configuración, al igual que con buildroot, será almacenada en un archivo llamado ".config" en el directorio principal del kernel de Linux y en base al cual podremos llevar a cabo la compilación.

La herramienta menuconfig proporciona una ayuda excelente sensible al contexto para cada entrada. A la derecha de cada entrada está un botón Help. Este es para desplegar una explicación detallada sobre qué hace la característica y por qué la incluiría directamente o como un módulo, o incluso por qué la excluiría. Cuando se tenga duda acerca de una característica, siempre utilice el botón Help para saber exactamente qué hace y por qué querría utilizarla.

A continuación describiremos algunas de las secciones que aparecen durante la configuración:

- **Loadable Module Support** En casi todos los casos, debes asegurarte de que el kernel es posible cargar módulos. Loadable Module Support despliega una lista de varias opciones de administración de módulos. Asegúrate de que Enable Loadable Module Support esté marcado en Yes. Esta característica le permite a tu kernel agregar módulos según sea necesario. Kernel Module Loader también debe estar en Yes, porque esto permite a sus daemon, como su servidor Web, cargar cualquier módulo que necesite.

- **Processor Type And Features** La ventana Processor Type And Features te permite configurar soporte a tu sistema particular. Aquí, se selecciona el tipo de procesador que tiene s (486, 586, 686, Pentium III, Pentium IV, etc.), además de la cantidad máxima de memoria en su soporte de sistema (hasta 64 GB con el kernel 2.4).

- **General Setup** La ventana General Setup te permite seleccionar características generales, como red, soporte a PCI en BIOS y administración de energía, además del soporte a ELF y binario a.out. También se da soporte a sysctl para cambiar de forma dinámica parámetros de kernel especificados en los archivos /proc. Se utiliza redhat-config-proc (la herramienta Kernel Tunning en el menú Herramientas del sistema) para hacer estos cambios dinámicos al kernel. En el menú de soporte de controlador de dispositivo adicional, se habilitan características especializadas como Crypto IP Encapsulation (CIPE) y SSL acelerado.

- **Block Devices (Device Drivers)** La ventana Block Devices incluye entradas que habilitan el soporte a los dispositivos IDE, unidades de disco flexible y dispositivos de puerto paralelos. Las características especiales, como soporte a discos RAM y el dispositivo para montar archivos de imagen de CD-ROM, también están ahí.

- **Multi-Device Support (RAID y LVM) (Device Drivers)** La ventana Multi-Device Support presenta entradas que habilitan el uso de dispositivos RAID. Se Puede seleccionar el nivel de soporte RAID que se quiera. Aquí también puede habilitar el soporte a Logical Volumen Management (LVM), que permite combinar particiones en volúmenes lógicos que se administran de forma dinámica.

- **Networking Options (Device Drivers/Networking Support)** La ventana Networking Options muestra un conjunto extenso de capacidades de red. La entrada TCP/IP Networking debe estar habilitada para permitir cualquier tipo de red de Internet. Aquí, se especifican características que permiten al sistema operar como una puerta de enlace, una firewall o un enrutador. Network Packet Filtering habilita el soporte a firewall Iptables. También existe el soporte a otros tipos de redes, incluidas AppleTalk e IPX. Apple Talk debe estar habilitado si se quiere utilizar NetTalk para conectarse a un sistema Macintosh en nuestra red (Filesystems).

- **ATA/IDE/MFM/RLL Support (Device Drivers)** En la ventana ATA/IDE/MFM/RLL Support, se puede seleccionar IDE, ATA and ATAPI Block Device para abrir una ventana donde se selecciona el soporte a discos duros ATA IDE y CD-ROM ATAPI.

- **SCSI Support (Device Drivers)** Si se tiene cualquier dispositivo SCSI en el sistema, hay que aegurarse de que las entradas en la ventana SCSI Support estén establecidas en Yes. Aquí se habilita el soporte a discos SCSI, unidades de cinta y CD-ROM. La ventana SCSI Low-Level Drivers despliega una lista extensa de dispositivos SCSI compatibles con Linux.

- **Network Device Support (Device Drivers/Networking Support)** La ventana Network Device Support presenta una lista de varias características generales para soporte a dispositivo de red. Aquí existen entradas para ventanas que incluyen soporte a tipos particulares de dispositivos de red, incluidos Ethernet y token ring, además de interfaces WAN y dispositivos AppleTalk. Muchos de estos dispositivos se crean como módulos que se cargan cuando se necesiten. Selecciona que se reconstruya el kernel con soporte a cualquiera de estos dispositivos generados directamente en el kernel.

- **Multimedia Devices (Device Drivers)** Los dispositivos multimedia proporcionan soporte a varias tarjetas multimedia, al igual que Video4Linux.

- **File Systems** En esta ventana se presentan los diferentes tipos de sistemas de archivos que Linux soporta. Éstos incluyen sistemas de archivos de Windows como DOS, VFAT y NTFS, además de archivos de CD-ROM como ISO y UDF. Se incluyen sistemas de archivos de red como NFS, SMB (Samba) y NCP (NetWare), además de varios sistemas de archivos como HFS (Macintosh).

- **Character Devices (Device Drivers)** La ventana Character Devices presenta características de dispositivos como teclado, ratón y puertos seriales. Existe soporte a ratón serial y de bus.

- **Sound (Device Drivers)** Para el kernel 2.4, la ventana Sound presenta una lista de diferentes tarjetas de sonido compatibles con el kernel. En el caso de sistemas más viejos, tal vez se tenga que proporcionar IRQ, DMA y E/S base que usa la tarjeta de sonido. Éstas se compilan como módulos separados, entre los que se selecciona si quiere que se incluyan directamente en el kernel. Para el kernel 2.6, se selecciona el soporte de sonido Advanced Linux Sound Architecture, que lo expande a controladores para dispositivos de sonido particulares (también se incluye Open Sound System, aunque esté descontinuado).

- **Bluetooth Devices (Device Drivers/Networking Support)** Aquí hay soporte a periféricos compatibles con Bluetooth, al presentar una lista de controladores para interfaces USB, serial y tarjetas PC.

Una configuración básica se muestra en las siguientes lineas:

Ejemplo de configuración del kernel por medio de menuconfig.

```
Processor support —>
Processor Type (AMCC 40x)
General setup —>
[ ] Support for paging of anonymous memory (swap)
-*- Configure standard kernel features (expert users) —>
[ ] Sysctl syscall support
[ ] Support initial ramdisks compressed using bzip2
[ ] Support initial ramdisks compressed using LZMA
[*] Embedded system
Platform support —>
<>Walnut
<*>Generic Xilinx Virtex board
[ ] KVM Guest support
Kernel options —>
Timer frequency (250 HZ)
[*] Math emulation
<* >Default bootloader kernel arguments
(console=ttyS0,9600 console=tty0 root=/dev/sda2) Initial kernel
(simpleImage.virtex405-nombre simpleImage.initrd.virtex405-nombre) Additional default images
[] Hibernation (aka 'suspend to disk')
Bus options —>
<>PCI support
<>PCCard (PCMCIA/CardBus) support
Networking support:
Networking options:
[*]IP: kernel level autoconfiguration: Yes (to enable NFS root)
<*>IP: DHCP support
<*>IP: BOOTP support
<*>IP: RARP support
[ ] Amateur Radio support —>
< > CAN bus subsystem support —>
< > IrDA (infrared) subsystem support —>
< > Bluetooth subsystem support —>
Wireless
< > cfg80211 - wireless configuration API
< > WiMAX Wireless Broadband support —>
< > RF switch subsystem support —>
< > Plan 9 Resource Sharing Support (9P2000) —>
Device drivers —>
{*} Connector - unified userspace <-> kernelspace linker —>
<*> Memory Technology Device (MTD) support —>
< > Parallel port support —>
<*>Block devices —>
< > Normal floppy disk support
<*> Loopback device support
<*> Network block device support
<*> RAM block device support
< > Packet writing on CD/DVD media
< > ATA over Ethernet support
<*>Xilinx SystemACE support
< > Virtio block driver (EXPERIMENTAL)
[*] Misc devices —>
< > Integrated Circuits ICS932S401
< > Enclosure Services
< > Intersil ISL29003 ambient light sensor
< > Taos TSL2550 ambient light sensor
< > Dallas DS1682 Total Elapsed Time Recorder with Alarm
< > Silicon Labs C2 port support (EXPERIMENTAL)
< > ATA/ATAPI/MFM/RLL support (DEPRECATED) —>
SCSI device support —>
[*] SCSI low-level drivers —>
<M> iSCSI Boot Sysfs Interface
< > Serial ATA and Parallel ATA drivers —>
<*>Network device support —>
[ ] ATM drivers —>
[*] Ethernet driver support (NEW) —>
[*] Xilinx devices
<*> Xilinx 10/100 Ethernet Lite support
<*> Xilinx LL TEMAC (LocalLink Tri-mode Ethernet MAC) driver
QUITAR TODO LO DEMAS
[ ] Wireless LAN —>
[ ] Wan interfaces support —>
[ ] ISDN support
< > Telephony support —>
Input device support —>
Hardware I/O ports —>
< > i8042 PC Keyboard controller
Character devices —>
[*] Legacy (BSD) PTY support
[ ] Non-standard serial port support
Serial drivers —>
<*>Xilinx uartlite serial port support
<*>Support for console on Xilinx uartlite serial port
<*> Xilinx PS UART support
[*] Xilinx PS UART console support
<*>Xilinx HWICAP Support
<*>GPIO Support —>
<*>Xilinx GPIO support
```

```
<> Multimedia support }
Graphics support —>
Console display driver support —>
[*] Map the console to the primary display device
<> Sound card support
<> MMC/SD/SDIO card support
<> Sony MemoryStick card support
[] LED Support
[] Accessibility support
File systems —>
<*>Second extended fs support
<*>Ext2 execute in place support
< > Ext3 journalling file system support
< > The Extended 4 (ext4) filesystem
< > Reiserfs support
< > JFS filesystem support
< > XFS filesystem support
< > GFS2 file system support
< > OCFS2 file system support
< > Btrfs filesystem (EXPERIMENTAL) Unstable disk format
< > NILFS2 file system support (EXPERIMENTAL)
<*>Network File Systems —>
<*>NFS client support
<*>Root file system on NFS
Library routines —>
<*> XZ decompression support
[*] Averaging functions
Kernel hacking —>
[ ] Debug memory initialisation
[ ] Virtualization —>
```

La configuración completa utilizada para este proyecto se anexa al final de este reporte en el apéndice.

3.4.3. Compilación

3.4.3.1. Usando initrd

Una vez concluida con la configuración, iniciamos la compilación del kernel con el siguiente comando:

: ∼/proyecto/linux-xlnx$ make simpleImage.initrd.virtex405-nombre

Esto sera para compilar el kernel junto con initial ramdisk (initrd) lo cual o discutimos en el tema de Sistema de Archivos. Creará además de vmlinux y System.map el archivo simpleImage.initrd.virtex405-nombre.elf. Una vez finalizada la compilación copiamos el archivo simpleImage.initrd.virtex405- nombre.elf a nuestro proyecto.

: ∼/proyecto/linux-xlnx$ cd arch/powerpc/boot/

: ∼/proyecto/linux-xlnx$ cp simpleImage.initrd.virtex405-nombre.elf

$\sim$/proyecto/xupv2p

La tarjeta XUPV2P es capaz de cargar un archivo binario en formato ELF (Execuable and Linkable Format) por medio de system.ace o por XMD, a continuación se describe el procedimiento para SysAce.

Necesitaremos una archivo de configuración (Archivo: xupGenace.opt). El archivo de configuración lucirá como sigue:

-jprog
-board user
-target ppc_hw
-hw implementation/download.bit

-elf simpleImage.initrd.virtex405-nombre.elf
-configdevice devicenr 1 idcode 0x1127e093 irlength 14 partname xc2vp30
-debugdevice devicenr 1 cpunr 1
-ace system.ace
Se genera como sigue:

: ~/proyecto/linux-xlnx$ cd ../xupv2p

: ~/proyecto/xupv2p$ xmd -tcl genace.tcl -opt xupGenace.opt

Esto nos generará el archivo system.ace. El archivo generado se coloca en la partición principal de la tarjeta compact flash que deberá tener formato fat16.

: ~/proyecto$ sudo mount /dev/DispositivoCompactFlash1 /mnt

: ~/proyecto$ cp $\sim$/proyecto/xupv2p/system.ace /mnt

: ~/proyecto$ umount /mnt

Habiendo hecho esto estaremos listos para ejecutar nuestro sistema en la tarjeta. Antes de ejecutar necesitamos iniciar minicom:

: ~/proyecto$ sudo minicom

Con esto podremos ver un kernel funcional ejecutándose sobre la memoria externa de la tarjeta.

3.4.3.2. Usando sistema de archivos

Para usar el sistema de archivos generado por Buildroot tendremos que volver a compilar nuestro kernel pero antes deberemos modificar el KLC en el device-tree para que busque el RFS en la Compact Flash:

: ~/proyecto$ cd inux-xlnx/arch/powerpc/boot/dts/

: ~/proyecto$ gedit virtex405-nombre.dts

En este archivo modificaremos la linea:

bootargs = "console=ttyUL0,115200 root=/dev/ram rw ip=dhcp";

por:

bootargs = "console=ttyUL0,115200 root=/dev/xs‘2 rw ip=dhcp";

Una vez hecho el cambio procedemos a compilar de nuevo el kernel pero ahora para el sistema de archivos:

: ~/proyecto/linux-xlnx$ make simpleImage.virtex405-nombre

: ~/proyecto/linux-xlnx$ cd arch/powerpc/boot/

: ~/proyecto/linux-xlnx$ cp simpleImage.virtex405-nombre.elf $\sim$/proyecto/xupv2p

Tendremos que modificar el archivo xupGenace.opt de la siguiente forma:
-jprog
-board user
-target ppc_hw
-hw implementation/download.bit
-elf simpleImage.virtex405-nombre.elf
-configdevice devicenr 1 idcode 0x1127e093 irlength 14 partname xc2vp30
-debugdevice devicenr 1 cpunr 1
-ace system.ace
Y generamos el archivo system.ace:

: ~/proyecto/linux-xlnx$ cd ../xupv2p

: ~/proyecto/xupv2p$ xmd -tcl genace.tcl -opt xupGenace.opt

Esto nos generará el archivo system.ace. El archivo generado se coloca en la partición principal de la tarjeta compact flash que deberá tener formato fat16.

: ~/proyecto$ sudo mount /dev/DispositivoCompactFlash1 /mnt

: ~/proyecto$ cp $\sim$/proyecto/xupv2p/system.ace /mnt

: ~/proyecto$ umount /mnt

Ya habiendo terminado con esto copiaremos y descomprimiremos el sistema de archivos generado por buildroot.

Ext2 es el sistema de archivos principal usado en este proyecto. Necesitaremos descomprimir el sistema de archivos generado por buildroot en una partición ext2 creada con anterioridad con fdisk en la tarjeta Compact Flash.

: ~/proyecto$ sudo mount /dev/DispositivoCompactFlash2 /mnt

: ~/proyecto$ cp $\sim$/proyecto/buildroot-2013.02/output/images/rootfs.tar.gz /mnt

: ~/proyecto$ cd /mnt

: ~/proyecto$ tar -zxf rootfs.tar.gz

: ~/proyecto$ umount /mnt

Habiendo hecho esto estaremos listos para ejecutar nuestro sistema en la tarjeta. Antes de ejecutar necesitamos iniciar minicom desde la terminal:

: ~/proyecto$ sudo minicom

Capítulo 4

Seguridad en redes

4.1. Introducción

En la actualidad son ampliamente utilizadas las redes de computadoras en muchos ámbitos, se implementan desde redes de área local, hasta redes de área amplia y todas estas tienen como finalidad compartir información, recursos u ofrecer servicios. Todo esto se da por medio de un proceso de comunicación donde existe una fuente, un transmisor, un sistema de transmisión, un receptor y un destino. Las organizaciones o usuarios que forman parte de estas redes están continuamente manejando e intercambiando información, esto genera la necesidad de protegerla de amenazas que podrían alterarla durante su transmisión, además necesitamos garantizar que los datos transmitidos sean auténticos.

Una red puede estar amenazada básicamente de dos formas distintas: una la llamaremos amenaza pasiva, en la cual un usuario ajeno ataca e intenta obtener información relativa a una comunicación, y la que llamaremos amenaza activa, en la que, además de obtener información, se le hace alguna modificación a los datos transmitidos o se crea una transmisión falsa.

La tecnología esencial en todas las aplicaciones de seguridad en redes es el cifrado de información y de esta existen dos técnicas fundamentales: cifrado simétrico y el cifrado de clave pública. En este proyecto se busca ofrecer esta seguridad mediante la implementación de una capa de seguridad con OpenSSL que este presente en un sistema empotrado que pueda realizar tanto el cifrado como el descifrado de información y que permita el intercambio seguro de información entre un cliente y un servidor WEB.

4.2. Conceptos Básicos

4.2.1. Cifrado simétrico

El cifrado simétrico cifra y descifra los datos usando una sola clave. La clave y el mensaje en texto plano son pasados a el algoritmo de cifrado, produciendo

el texto cifrado. El resultado pued eser enviado por medio de un medio inseguro, permitiendo solo al destinatario, quien tiene la clave original, descifrar el mensaje, lo cual se lleva a cabo pasando el texto cifrado y la clave a el algoritmo de descifrado. Obviamente, la clave debe mantenerse secreta para que este esquema sea efectivo.

Aunque no son los únicos algoritmos de cifrado que existen, solo mencionaremos los dos más importantes, siendo estos dos cifradores de bloque, lo cual significa que se procesa un texto nativo en bloques de tamaño fijo y produce un bloque de texto cifrado de igual tamaño por cada bloque de texto nativo:

- **Estándar de cifrado de datos (DES)** En un principio utilizaba una clave de 56 bits y su vida se prolongo gracias a la creación de lo que se conoció como 3DES que solo es la repetición del mismo algoritmo 3 veces y con una longitud de clave ahora de 112 o 168 bits.

- **Estándar de cifrado avanzado(AES)** Surgió como el algoritmo que debía sustituir a DES. Este algoritmo mejora significativamente la eficiencia siendo igualmente un cifrador de bloque simétrico con una longitud de bloque de 128 bits y admite longitudes de clave de 128, 192 y 256 bits.

Cabe mencionar que para la distribución de claves se utiliza un Centro de Distribución de Claves(KDC,Key Distribution Center) el cuál determina a que equipos se les permite comunicarse entre ellos y les otorga una clave de sesión en nombre de los usuarios.

4.2.2. Autenticación de mensajes y funciones de dispersión (hash)

Para poder proteger nuestros mensajes de los ataques activos antes mencionados requerimos de lo que se conoce como autenticación del mensaje, lo cual nos asegura que le mensaje que se mando es exactamente el mismo que se esta recibiendo.

La autenticación de mensajes se puede dar creando un **Código de Autenticación de Mensaje** (MAC) que no es más que un pequeño bloque de datos que se adjunta al mensaje al momento del envío y se genera a partir del mensaje en texto plano y una clave secreta. Estos dos elementos son entradas para un algoritmo para crear el MAC. El destinatario realiza los mismos cálculos sobre el mensaje recibido utilizando la misma clave secreta y el resultado debe ser el mismo MAC que fue recibido junto con el mensaje.

Otra manera de hacerlo es mediante una **función de dispersión**. Esta función nos crea un resumen de longitud fija del mensaje que deseamos enviar y lo adjunta al mismo. Es importante resaltar que aquí no se utiliza una clave secreta como entrada. El destinatario, al recibir el mensaje, creara su propio resumen utilizando el mismo algoritmo y el texto nativo y lo deberá comparar con el resumen que esta recibiendo y estos deben ser exactamente iguales, si no es así, significará que el mensaje fue alterado.

El resumen del mensaje se puede cifrar usando cifrado simétrico o cifrado de clave publica para dar mayor seguridad. Además de estas dos técnicas existe también la posibilidad de usar un valor secreto. Este valor, que debe ser solo compartido entre el emisor y el destinatario, se usa junto con el texto para crear el resumen

4.2.3. Cifrado de clave pública

El cifrado de clave pública ofrece una solución al problema de tener que usar la distribución de claves como se hace en el cifrado simétrico. En este método cada parte tiene dos claves, una que debe mantenerse secreta (clave privada) y otra que debe ser libremente distribuida (clave pública). Las dos claves tienen una especial relación matemática que las une.

En la práctica las claves públicas son proporcionadas con un conjunto de información de soporte llamado certificado y los cuales son validados por una tercera persona de confianza. Frecuentemente esta tercera persona es una organización que investiga a la persona que desea tener sus certificados validados.

Debemos mencionar que aunque este método tiene sus ventajas sobre el cifrado simétrico tiene la gran desventaja de ser demasiado lento. Como resultado de esto, muchos de los sistemas que utilizan cifrado de clave pública, incluyendo SSL, lo usan tan poco como sea posible. Generalmente este método es usado para compartir y quedar de acuerdo en las claves que se utilizarán en un cifrado simétrico y entonces posteriormente todo el cifrado de información sea hecho usando este último método.

El algoritmo de clave pública más utilizado es RSA

4.2.3.1. Certificados

En términos simples, un certificado enlaza una clave pública con un nombre de identidad. Este nombre de identidad es simplemente el nombre de una persona o entidad que publicó, y por tanto, a la que le pertenece la clave pública a la cual está ligado el certificado. Un certificado es emitido por una Autoridad Certificadora (CA) y contiene información acerca del sujeto dueño de la clave pública así como de la misma autoridad que esta emitiendo el certificado. Contiene elementos de seguridad que permiten su autenticidad y evitan, por lo tanto falsificaciones. Un certificado es valido solo por un tiempo definido y este es firmado con la clave privada del emisor, es decir, por la autoridad certificadora.

Una CA es una organización o compañía que emite certificados y tiene la enorme responsabilidad de asegurar que los certificados emitidos son legítimos. Hay dos tipos básicos de CA:

- CA Privada. Solo para miembros de una misma organización
- CA Pública. Como Verisign o Thawte, tienen la responsabilidad de emitir certificados para cualquier persona y debe ser de confianza para todos los usuarios.

Entonces, para que el servidor, del que es objeto este proyecto, sea de confianza necesita un certificado que permita verificar que la clave pública que emite es realmente de él y así el cliente pueda tener la confianza de transmitir y recibir datos.

4.2.3.2. Obtener certificado

Generaremos nuestro propio certificado el cual será emitido por una CA privada que seremos nosotros mismos y para eso haremos uso de OpenSSL.

Primero debemos describir nuestra jerarquía de certificados. Tendremos solo un certificado de confianza del cuál se derivaran todos los certificados que queramos generar después y al cuál llamaremos root CA. Este root CA representará la CA para nuestra empresa. Para verificar la validación de nuestros certificados simplemente se tiene que hayan sido firmados por el root CA. También crearemos nuestra propia CA la cuál estará firmada por el root CA y será usada para firmar todos los certificados de identidad.

Primero crearemos el directorio que almacenara todos los archivos necesarios:

: ∼/proyecto/servidor$ mkdir claves

: ∼/proyecto/servidor$ cd claves

Ahora seguiremos los siguientes pasos:
1. Para crear el root CA:

: ∼/proyecto/servidor/claves$ openssl req -newkey rsa:1024 -sha1 -keyout rootkey.pem -out rootreq.pem

: ∼/proyecto/servidor/claves$ openssl x509 -days 3650 -req -in rootreq.pem -sha1 -extfile /etc/ssl/openssl.cnf -extensions v3_ca -signkey rootkey.pem -out rootcert.pem

: ∼/proyecto/servidor/claves$ cat rootcert.pem rootkey.pem > root.pem

2. Para crear la CA del servidor y firmarla con el root CA:

: ∼/proyecto/servidor/claves$ openssl req -newkey rsa:1024 -sha1 -keyout serverCAkey.pem -out serverCAreq.pem

: ∼/proyecto/servidor/claves$ openssl x509 -days 3650 -req -in serverCAreq.pem -sha1 -extfile /etc/ssl/openssl.cnf -extensions v3_ca -CA root.pem -CAkey root.pem -CAcreateserial -out serverCAcert.pem

: ∼/proyecto/servidor/claves$ cat serverCAcert.pem serverCAkey.pem rootcert.pem > serverCA.pem

3. Para crear el certificado del servidor y firmarlo con la CA del servidor:

: ∼/proyecto/servidor/claves$ openssl req -newkey rsa:1024 -sha1 -keyout serverkey.pem -out serverreq.pem

: ∼/proyecto/servidor/claves$ openssl x509 -days 3650 -req -in serverreq.pem -sha1 -extfile /etc/ssl/openssl.cnf -extensions usr_cert -CA serverCA.pem -CAkey serverCA.pem -CAcreateserial -out servercert.pem

: ∼/proyecto/servidor/claves$ cat servercert.pem serverkey.pem serverCAcert.pem rootcert.pem >> server.pem

: ∼/proyecto/servidor/claves$ openssl x509 -subject -issuer -noout -in server.pem

En cada paso deberás ingresar el pass phrase para la clave y después de esto deberás responder unas preguntas como se muestra en el ejemplo mas abajo. La parte más importante es la de "Common Name" que es en dónde se debe responder con el hostname con el cual los clientes se conectarán al servidor. Un ejemplo podrá ser "www.mydomain.com" y para este proyecto pondremos Root CA en el primer paso, Server CA en el segundo y solo localhost en el último paso. El último comando del paso 3 sirve para verificar los datos ingresados durante esta parte.

Al final tendremos varios archivos generados pero los que mas nos interesan son los archivos server.pem y root.pem los cuales contendrá tanto la clave privada como el certificado público basado en ella. El certificado será valido por 3650 días (10 años) y la clave estará cifrada. Opcionalmente podemos agregar la opción **-nodes** para que la clave no sea cifrada y cabe mencionar que la opción **-extfile** hace referencia a la ruta donde tenemos ubicado nuestro archivo de configuración.

Ejemplo de creación de claves y certificado

```
: ~/proyecto/servidor/claves$ openssl req -newkey rsa:1024 -sha1 -keyout rootkey.pem -out rootreq.pem
Generating a 1024 bit RSA private key
..............++++++
.....................++++++
writing new private key to 'rootkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
---
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
---
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:Mexico
Locality Name (eg, city) []:Mexico
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Antares, Inc
Organizational Unit Name (eg, section) []:Development
Common Name (eg, YOUR name) []:Root CA
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Ahora solo nos queda copiar los archivos server.pem y root.pem a la carpeta que contendrá nuestro servidor:

: ~/proyecto/servidor/claves$ cp server.pem root.pem ../

4.2.3.3. Diffie-Hellman

Diffie-Hellman es usado para establecer el acuerdo entre claves. El acuerdo entre claves es el intercambio de información sobre un medio no seguro que permite a cada una de las do partes en una conversación computar un valor que es generalmente usado como una clave para un cifrador simétrico.

Lo primero que hace es crear un conjunto de parámetros que son aceptados por ambas partes en la conversación. Los parámetros, que consisten de un número primo aleatorio y un generador, son públicos y pueden ser aceptados ya sea antes de que la conversación comience o intercambiados como parte de la conversación. Ya aceptados los parámetros, son usados para que cada parte compute una clave pública y una privada. Posteriormente las dos partes intercambian sus claves públicas y así estarán listas para el intercambio de información.

Para nuestro servidor es necesario crear un archivo que contenga estos parámetros y lo hacemos de la siguiente forma:

: ~/proyecto/servidor$ openssl dhparam -outform PEM -out dhparam.pem -2 1024

Esto nos genera un conjunto de parámetros Diffie-Hellman usando un generador de 2, un número primo aleatorio de 1024 bits y escribe los parámetros en el formato PEM en el archivo dhparam.pem.

4.2.4. Firmas digitales

Una firma digital es necesaria cuando se requiere tener la certeza de que un mensaje proviene realmente de la fuente que dice provenir. Para esto, el emisor utiliza su propia clave privada para cifrar el mensaje. Cuando el destinatario recibe el texto cifrado, comprueba que puede descifrarlo con la clave pública del emisor, demostrando así que el mensaje ha tenido que ser cifrado por el emisor en cuestión y no por nadie mas. De esta forma todo el mensaje cifrado sirve como firma digital. Además es imposible alterar el mensaje sin tener acceso a la clave privada por lo que el mensaje esta autenticado en términos de origen e integridad de los datos.

Es importante enfatizar que la firma digital no ofrece privacidad. Es decir, el mensaje que se envía esta seguro frente a alteraciones, pero no lo esta de ser leído por otros.

4.3. SSL y TLS

Uno de los servicios de seguridad más ampliamente utilizados es el de capa de sockets segura (SSL) y el posterior estándar de Internet conocido como capa de transporte segura (TLS), definido este último en el RFC 2246. SSL es un servicio de propósito general implementado como un conjunto de protocolos que hacen uso de TCP. En este nivel, existen dos elecciones de implementación. Para una completa generalidad, SSL (o TLS) podría suministrarse como parte de la familia de protocolos subyacente y, de esta forma, ser transparente a las aplicaciones. Alternativamente, SSL puede integrarse en paquetes específicos. Por ejemplo, los navegadores Netscape y Microsoft Explorer vienen equipados con SSL y la mayoría de los servidores web implementan este protocolo.

En este proyecto se discute SSLv3. En TLS sólo existen modificaciones menores.

4.3.1. Arquitectura SSL

SSL hace que las conexiones de red sean seguras y esto funciona bien con TCP/IP con objeto de proporcionar un servicio fiable y seguro extremo a extremo. Sin embargo, no funciona con todas las capas de protocolos que no son orientados a conexión, tales como UDP y IPX. Veamos como es que funciona el protocolo.

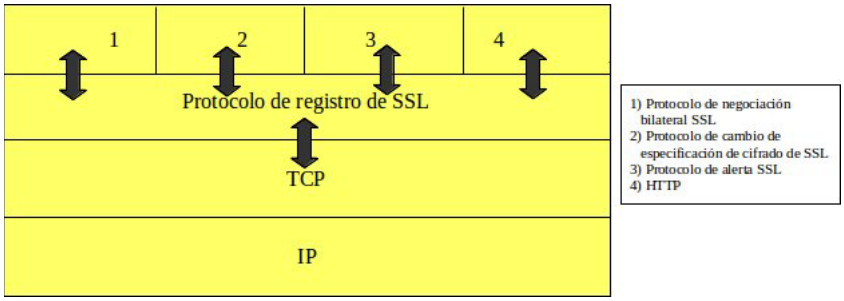

Figura 4.1: Arquitectura SSL

El protocolo de registro de SSL proporciona:

- Privacidad ya que establece una clave secreta compartida que es utilizada para el cifrado simétrico.
- Integridad ya que define una clave secreta para formar el MAC

4.3.1.1. Protocolo de negociación bilateral

Permite al servidor y al cliente autenticarse mutuamente para negociar algoritmos de cifrado y de MAC y de claves criptografías que se usarán para proteger los datos enviados.

- Fase 1: Establecer capacidades de seguridad, incluyendo versión del protocolo, identificador de sesión, repertorio de cifrado, método de compresión y números aleatorios iniciales.
- Fase 2: El servidor debe enviar el certificado, intercambiar las claves y solicitar certificados. El servidor indica el final de la fase de saludo.
- Fase 3: El cliente envía su certificado si se el ha solicitado. El cliente envía intercambio de clave. El cliente puede enviar la verificación del certificado.
- Fase 4: Cambiar repertorio de cifrado y finalizar protocolo de negociación bilateral.

4.4. OpenSSL

4.4.1. ¿Qué es OpenSSL?

OpenSSL es un esfuerzo colaborativo para desarrollar una implementación robusta, de grado comercial, con todas las características y herramientas de código abierto que están presentes en los protocolos SSL(Secure Sockets Layer) y TLS(Transport Layer Security) así como una librería de criptografía de propósito general robusta. El proyecto es gestionado por una comunidad mundial de voluntarios que utilizan Internet para comunicarse, planear y desarrollar el kit de herramientas OpenSSL y la documentación relacionada.

OpenSSL esta basada en la librería SSLeay desarrollada por Eric A. Young y Tim J. Hudson y es esencialmente dos herramientas en una: una librería de cifrado y un conjunto de herramientas de SSL.

La librería SSL provee una implementación de todas las versiones del protocolo SSL incluyendo TLSv1. La librería de cifrado nos ofrece los más populares algoritmos para cifrado simétrico y de clave pública, algoritmos hash y para resumen de mensajes. Incluye también un generador de números pseudoaleatorios y soporte para la manipulación de formatos comunes de certificados y material para el manejo de claves.

4.4.2. Instalación

Podemos obtener OpenSSL Directamente desde la página http://www.openssl.org/:

: ∼/proyecto$ wget https://www.openssl.org/source/openssl-0.9.7c.tar.gz

Ya que lo tenemos los pasos para su instalación son los siguientes:

: ∼/proyecto$ tar -xvf openssl-0.9.7c.tar.gz

: ∼/proyecto$ cd openssl-0.9.7c

: ∼/proyecto$ ./config

: ∼/proyecto$ make test # Este paso es opcional

: ∼/proyecto$ su # Necesitas ser superusuario para instalar

: ∼/proyecto$ make install

El procedimiento anterior es para ilustrar la instalación de OpenSSL en un sistema Linux mas sin embargo la mayoría de las distribuciones ya vienen con OpenSSL por defecto así que no tendremos que preocuparnos por tener que instalarlo en casi ninguna situación.

Para este proyecto tampoco es necesario realizar el procedimiento anterior ya que como explicamos en el capitulo 3 *Sistema de archivos* nosotros creamos el sistema de archivos con la configuración apropiada para que OpenSSL ya estuviera cargado en nuestro sistema. Como ya mencionamos, el procedimiento aquí expuesto es solo ilustrativo.

4.4.3. Configuración

OpenSSL ofrece mediante su herramienta de linea de comandos una extensa lista de comandos y varias opciones para cada una de estos. Poder aprender y utilizar toda esta amplia variedad de comandos puede ser algo complicado y frustrante a la vez y es por eso que debemos hacer uso de un archivo de configuración que simplifique nuestro trabajo.

En nuestra instalación ya viene incluido un archivo de configuración con los ajustes básicos para un buen funcionamiento aunque siempre tenemos abierta la posibilidad de modificar alguno de sus parámetros para ajustarlo a nuestras necesidades. Para encontrar este archivo de configuración primero debemos de ingresar a la herramienta de linea de comandos de OpenSSL e ingresar al comando ca:

: ∼/proyecto/servidor$ openssl

OpenSSL> ca

OpenSSL> quit # para salir de la utilidad

Esto nos dará la ubicación del archivo de configuración y podría ser que mostrará algunos errores los cuales pueden ser ignorados ya que no representan ningún problema. Realmente no todos los comandos hacen uso del archivo de configuración pero los que si lo hacen principalmente son tres: **ca, req, x509**.

Parte de código de archivo de configuración

```
[ ca ]
default_ca = CA_default # The default ca section
####################################################################
[ CA_default ]
dir = ./demoCA # Where everything is kept
certs = $dir/certs # Where the issued certs are kept
crl_dir = $dir/crl # Where the issued crl are kept
database = $dir/index.txt # database index file.
#unique_subject = no # Set to 'no' to allow creation of
# several ctificates with same subject.
new_certs_dir = $dir/newcerts # default place for new certs.
certificate = $dir/cacert.pem # The CA certificate
serial = $dir/serial # The current serial number
crlnumber = $dir/crlnumber # the current crl number
# must be commented out to leave a V1 CRL
crl = $dir/crl.pem # The current CRL
private_key = $dir/private/cakey.pem# The private key
RANDFILE = $dir/private/.rand # private random number file
```

```
x509_extensions = usr_cert # The extentions to add to the cert
# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt = ca_default # Subject Name options
cert_opt = ca_default # Certificate field options
# Extension copying option: use with caution.
# copy_extensions = copy
# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions = crl_ext
default_days = 365 # how long to certify for
default_crl_days= 30 # how long before next CRL
default_md = sha1 # which md to use.
preserve = no # keep passed DN ordering
```

El archivo de configuración completo utilizado en este proyecto se muestra en el apéndice al final de este documento.

Capítulo 5

Servidor HTTPS

5.1. Introducción

El término ordenador local se utiliza para referirse al ordenador que el usuario utiliza para entrar en la red Internet. Desde ese ordenador el usuario establece conexiones con otros ordenadores, denominados ordenadores remotos, a los que solicita algún servicio. Estos ordenadores remotos que ofrecen servicios reciben también el nombre de servidores o host.

La utilización de las diferentes aplicaciones o servicios de Internet se lleva a cabo respondiendo al llamado modelo cliente-servidor.

Cuando se utiliza un servicio en Internet, como consultar una base de datos, transferir un fichero o participar en un foro de discusión, se establece un proceso en el que entran en juego dos partes. Por un lado, el usuario, quien ejecuta una aplicación en el ordenador local: el denominado programa cliente. Este programa cliente se encarga de ponerse en contacto con el ordenador remoto para solicitar el servicio deseado. El ordenador remoto por su parte responderá a lo solicitado mediante un programa que está ejecutando. Este último se denomina programa servidor. Los términos cliente y servidor se utilizan tanto para referirse a los programas que cumplen estas funciones, como a los ordenadores donde son ejecutados esos programas.

Ahora bien, cuando se requiere que esta comunicación sea segura, es decir, cuando se quiere tener la certeza de que la información que se envía y recibe a través de la red sea vista solo por el emisor y el receptor del mensaje y aunque alguien pudiera interceptar el mensaje enviado este sea incapaz de ver el contenido, entonces se debe agregar una capa de seguridad que permita esta confidencialidad. En este capitulo se describirá como realizar esta tarea, primero explicando como funciona el modelo cliente-servidor, un servidor que no utiliza cifrado y después se explicará como agregarle una capa de seguridad haciendo uso de OpenSSL.

5.2. Modelo cliente-servidor

El modelo cliente-servidor es uno de los más empleados para construir aplicaciones en una red. Un servidor es un proceso que se esta ejecutando en un nodo de la red y que gestiona el acceso a un determinado recurso. Un cliente es un proceso que se ejecuta en el mismo o diferente nodo y que realiza peticiones al servidor. Las peticiones están originadas por la necesidad de acceder al recurso que gestiona el servidor.

El servidor está continuamente esperando peticiones de servicio. Cuando se produce una petición, el servidor despierta y atiende al cliente. Cuando el servicio concluye, el servidor vuelve al estado de espera. De acuerdo a la forma de prestar el servicio, podemos considerar dos tipos de servidores:

- *Servidores interactivos.* El servidor no solo recoge la petición del servicio, sino que el mismo se encarga de atenderla. }
- *Servidores concurrentes.* El servidor recoge cada una de las peticiones de servicio y crea otros procesos para que se encarguen de atenderlas.

Para nuestro proyecto iniciaremos creando un servidor interactivo pero le agregaremos los elementos necesarios para convertirlo en un servidor concurrente. Esto dará un mejor rendimiento a nuestro servidor.

5.2.1. Esquema general de un servidor y un cliente-servidor

Las acciones que debe llevar a cabo el programa servidor son las siguientes:

1. Abrir el canal de comunicaciones e informar a la red tanto de la dirección a la que responderá como de su disposición para aceptar peticiones de servicio.
2. Esperar a que un cliente le pida servicio en la dirección que él tiene declarada.
3. Cuando recibe una petición de servicio, si es un servidor interactivo, atenderá al cliente. Si el servidor es concurrente, creará un proceso mediante **fork** para que le de servicio al cliente.
4. Volver al punto 2 para esperar nuevas peticiones de servicio.

El programa cliente, por su parte, llevará a cabo las siguientes acciones:

1. Abrir el canal de comunicaciones y conectarse a la red atendida por el servidor. Naturalmente , esta dirección debe ser conocida por el cliente y responderá al esquema de generación de direcciones de la familia de conectores que se este empleando.
2. Enviar al servidor un mensaje de petición de servicio y esperar hasta recibir la respuesta.
3. Cerrar el canal de comunicaciones y terminar la ejecución.

5.3. Manejo de sockets

5.3.1. Llamadas para le manejo de conectores

A continuación veremos cada una de las llamadas al sistema que intervienen en la codificación de programas servidores y clientes que se comunican mediante conectores. Mostraremos esto para entender la base de nuestro servidor y como es que este lleva acabo el envío y recepción de información. Hay que tener en cuenta que nuestro servidor se programó utilizando el lenguaje de programación C y mostraremos pequeños fragmentos de código para ejemplificar cada parte. El código completo del servidor esta al final en el apéndice.

5.3.1.1. Apertura de un punto terminal en un canal (socket)

La llamada para abrir un canal bidireccional de comunicación es **socket** y se declara como sigue:

#include <sys/types.h>
#include <sys/socket.h>
int socket (int af, int type, int protocol);

La llamada crea un punto terminal para conectarse a un canal y devuelve un descripto. El descriptor del conector devuelto se usará en llamadas posteriores a funciones de la interfaz. El parámetro **af** (address family) especifica la familia de conectores o familia de direcciones que se desea emplear.

Las distintas familias están definidas en el fichero de cabecera **<sys/socket.h>** y las dos principales son:

- AF_UNIX, protocolos internos UNIX. Es la familia de conectores empleada para comunicar procesos que se ejecutan en una misma máquina.
- AF_INET, protocolos Internet. Es la familia de conectores que se comunican mediante protocolos tales como TCP (Transmission Control Protocol).

Como nuestro servidor deberá ser capaz de recibir peticiones desde nodos conectados a nuestra red red local tendremos que utilizar AF_UNIX.

El parámetro **type** indica la semántica de la comunicación para el conector, los dos principales valores que puede tomar:

- SOCK_STREAM, conector con un protocolo orientado a conexión.
- SOCK_DGRAM, conector con un protocolo no orientado a conexión o datagrama.

Nuestro servidor utilizará un conector con un protocolo orientado a conexión.

El parámetro **protocol** especifica el protocolo particular que se va a usar con el conector. Si toma un valor de 0 la elección del protocolo se deja en manos del sistema.

5.3.1.2. Nombre de un conector (bind)

La llamada **bind** se utiliza para unir un conector con una dirección de red determinada, su declaración es la siguiente:

#include <sys/socket.h>
#include <sys/un.h> //Solo para la familia AF_UNIX
#include <sys/netinet.h> //Solo para la familia AF_INET
int bind (int sfd, const void *addr, int addrlen);

Cuando se crea un conector con la llamada **socket**, se le asigna una familia de direcciones, pero no una dirección particular, **bind** hace que el conector cuyo descriptor es **sfd** se una a la dirección de conector especificada en la estructura apuntada por **addr**, **addrlen** indica el tamaño de la dirección. Para AF_UNIX se debe usar la estructura **struct socckaddr_un** y para la familia AF_INET la estructura **struct sockaddr_in**. El valor de addrlen se puede calcular con el operador **sizeof**.

Si se une un conector AF_INET a una dirección, el campo **sin_port** puede contener el número de un puerto o 0. Si vale 0, el sistema le asigna el número de un puerto libre.

5.3.1.3. Disponibilidad para recibir peticiones de servicio (listen)

Cuando se abre un conector orientado a conexión el programa servidor indica que esta disponible para recibir peticiones de conexiones mediante la llamada **listen**, que se declara como sigue:

int listen (int sfd, int backlog);

La llamada listen la suele ejecutar el proceso servidor después de las llamadas a **socket** y **bind**; **listen** habilita una cola asociada al conector descrito por **sfd**, esta cola se utiliza para alojar peticiones de conexiones procedentes de los procesos cliente. Para que la llamada a **listen** tenga sentido, el conector debe ser del tipo SOCK_STREAM.

La cola de conexiones es importante para los servidores de tipo interactivo, porque mientras están atendiendo a un cliente pueden llegar peticiones procedentes de otros. En los de tipo concurrente, el único instante en el que el servidor no atiende la recepción de peticiones es el que dedica a la llamada **fork**. En estos casos también es importante la cola de conexiones.

5.3.1.4. Petición de conexión (connect)

Para que un proceso cliente establezca una conexión con un servidor es necesario que haga una llamada a **connect**. Esta función se declara de la forma siguiente:

int connect (int sfd, const void *addr, int addrlen);

donde **sfd** es el descriptor del conector que da acceso al canal, **addr** es un puntero a una estructura que contiene la dirección del conector remoto al que queremos conectarnos y **addrlenes** el tamaño en bytes de la dirección. La estructura de la dirección dependerá de la familia de conectores con la que estemos trabajando.

5.3.1.5. Aceptación de una conexión

Los procesos servidores podrán leer peticiones de servicio mediante la llamada **accept**:

int accept (int sfd, void *addr, int *addrlen);

Esta llamada se usa con conectores orientados a conexión. El argumento **sfd** es un descriptor del conector creado por una llamada previa a **socket** y unido a una dirección mediante **blind**. La llamada **accept** extrae la primera petición de conexión que hay en la cola de peticiones pendientes, creada con una llamada a **listen**. Una vez extraída la petición, **accept** crea un nuevo conector con las mismas propiedades que **sfdy** reserva un nuevo descriptor de fichero **nfsd** para él.

El argumento **addr** debe apuntar a una estructura local con la dirección del conector. La llamada **accept** rellenará esta estructura con la dirección del conector remoto que pide la conexión. La función sobrescribirá en **addrlen** el tamaño real de la dirección leída de la cola de direcciones.

5.3.1.6. Lectura o recepción y envió de mensajes

En el programa básico del servidor se trata al conector como si fuera un fichero, por lo tanto, para enviar y recibir información haremos uso de la función **fdopen** la cual asocia un flujo con un descriptor:

fdopen(int file, const char *mode)

le daremos como argumento el descriptor del conector que obtuvimos con la llamada a **accept** y a+ para leer y añadir sobre este.

5.3.1.7. Cierre del canal

Una vez que un proceso no necesita realizar más accesos a un conector, puede desconectarse del mismo. Para ello, aprovechando que un conector es tratado sintácticamente como si fuera un fichero, podemos usar la llamada **close**. Esta llamada cierra el conector en sus dos sentidos (servidor-cliente y cliente-servidor).

5.4. Capa de seguridad sobre el servidor

Una forma rápida y sencilla de hacer segura una conexión de red basada en el protocolo TCP es usando OpenSSL. Nosotros usaremos la API de OpenSSL para cubrir nuestro servidor con una capa de seguridad.

5.4.1. Inicializar nuestro contexto

Nuestra primer tarea será levantar un objeto de contexto (con SSL_CTX). Este objeto es usado para crear un nuevo objeto de conexión para cada nueva conexión SSL. Son esos objetos los que son usados para llevar a cabo el *handshake*, lecturas y escrituras.

Esto tiene dos ventajas. La primera es que el objeto de contexto permite que varias estructuras sean inicializadas solo una vez, mejorando el rendimiento. En muchas aplicaciones, cada conexión SSL usa el mismo conjunto de claves, lista de CA, etc. Para no tener que leer todo este material para cada conexión, nosotros simplemente cargamos todo esto en el objeto de contexto al inicio del programa. Cuando nosotros deseemos crear una nueva conexión, simplemente dirigimos esa conexión al objeto de contexto. La segunda ventaja es que permite múltiples conexiones SSL para compartir datos, tales como las sesiones SSL en cache usadas para la reanudación de sesión. Todo esto es llevado a cabo con la función **initialize_ctx()**.

5.4.2. Inicializar librerías y crear el contexto

Antes de que OpenSSL pueda ser usado debe ser inicializada la librería. Esto es llevado a cabo con **SSL_load_error_strings()**, la cual leerá los algoritmos que OpenSSL usará. Si queremos también un buen reporte de errores, necesitaremos leer las cadenas de errores usando **SSL_load_error_strings()**.

También creamos un objeto que es usado para mostrar los errores. OpenSSL usa una abstracción que es llamada un objeto **BIO** para la entrada y salida. Esto permite al programador usar las mismas funciones para diferentes tipos de canales de I/O (sockets, terminal, buffers de memoria, etc.) simplemente usando diferentes tipos de objetos **BIO**. En este caso crearemos un objeto **BIO** para stderr para imprimir los errores.

5.4.3. Leer nuestra claves

Para levar a cabo la autenticación necesitaremos leer nuestra clave pública y privada y el certificado asociado. El certificado es leído y cargado junto con los certificados de las CA formando la cadena de certificados usando **SSL_CTX_use_certificate_chain_file()**. Usaremos **SSL_CTX_use_PrivateKey_file()** para leer la clave privada. Por razones de seguridad, la clave privada es usualmente cifrada bajo password. Si la clave esta cifrada, el password será obtenido por medio de **SSL_CTX_set_default_passwd_cb()**.

5.4.4. Leer lista root CA

Para realizar la autenticación se necesita saber en que CAs confías. La llamada a **SSL_CTX_load_locations()** es usada para leer las CAs.

5.4.5. Leer el generador de números aleatorios

Para tener una buena seguridad, SSL necesita una buena fuente de números aleatorios. En general, es responsabilidad de la aplicación suministrar una semilla para el generador de números aleatorios. Sin embargo, OpenSSL automáticamente usa **/dev/urandom** para esto si esta disponible en el sistema. Este

generador es un estándar en Linux así que no tendremos problema alguno en esta parte ya que el generador es bastante sofisticado y fácil de ajustar.

5.5. El servidor

De manera general lo que hace nuestro servidor es, primero, usar la función **fork()** para así poder manejar múltiples clientes. Después usamos la API **BIO** de OpenSSL para leer la solicitud del cliente una linea a la vez así como escribir la respuesta correspondiente a el cliente. Finalmente, el servidor cierra la conexión.

5.5.1. Capa de abstracción

Una vez que se ha inicializado el contexto de SSL el servidor esta listo para recibir solicitudes del cliente. OpenSSL necesita hacer uso de un socket TCP, el cuál ya describimos como funciona al inicio de este capitulo, para ahora crear un socket SSL.

Una vez que ha sido creada una conexión TCP, creamos entonces un objeto SSL para manejar la conexión. Este objeto necesita ser unido al socket. Para esto creamos un objeto BIO usando el socket y entonces unimos el objeto SSL a el BIO.

Esta capa de abstracción permite usar OpenSSL también sobre otros canales que no sean sockets. Por ejemplo podríamos generar un programa que conecte a un cliente y a un servidor solo por medio de buffers o sobre una linea por puerto serial.

5.5.2. Accept y Fork

En Linux, la manera más simple para escribir un servidor que pueda manejar múltiples clientes es creando un nuevo proceso para cada cliente que se conecte y para eso llamamos a la función **fork()** después de **accept()**. Cada nuevo proceso se ejecuta independientemente y solo sale cuando termina de atender al cliente.

```
while(1){
if((s=accept(sock,0,0))<0)
err_exit("Problem accepting");
if((pid=fork())){
close(s);
}
else {
sbio=BIO_new_socket(s,BIO_NOCLOSE);
ssl=SSL_new(ctx);
SSL_set_bio(ssl,sbio,sbio);
if((r=SSL_accept(ssl)<=0))
berr_exit("SSL accept error");
http_serve(ssl,s);
exit(0);
}
}
```

5.5.3. Server Accept

Después de haber generado un nuevo proceso con **fork()** y haber creado el objeto SSL, el servidor llama a **SSL_accept()** lo que provoca que OpenSSL

lleve a cabo el *handshake*. La única situación en la que **SSL_accept()** terminará es cuando el *handshake* se haya completado o un error haya sido detectado.

Buffered I/O

Los objetos **BIO** de OpenSSL pueden usarse junto con otros objetos. Nosotros podemos envolver un objeto **SSL** en un objeto **BIO** y entonces envolver el objeto **BIO** en un buffer **BIO**.

io=BIO_new(BIO_f_buffer());
ssl_bio=BIO_new(BIO_f_ssl());
BIO_set_ssl(ssl_bio,ssl,BIO_CLOSE);
BIO_push(io,ssl_bio);

Esto permite hacer lecturas desde el buffer y escribirlas sobre la conexión SSL usando las funciones **BIO_*** sobre el nuevo objeto io. Esto hará más sencillo escribir el código ya que permite trabajar de una manera más natural al programador.

5.5.4. Request

Una solicitud (request) consiste de una linea de solicitud seguida de varias lineas de cabecera y opcionalmente un cuerpo. El final de las lineas de cabecera es indicada por una linea en blanco (un par de CRLFs). La manera más conveniente de leer la solicitud es leerla una linea a la vez hasta encontrar la linea en blanco. Nosotros podemos hacer esto usando la llamada a la función **BIO_gets()** de OpenSSL. Una vez que obtenemos la primer linea, la analizamos para así saber que es lo que nos esta solicitando el cliente.

```
r=BIO_gets(io,buf,BUFSIZZ-1);
printf(buf);
method = strtok(buf, " ");
path = strtok(NULL, " ");
protocol = strtok(NULL, "\r");
if (!method || !path || !protocol) return -1;
////////////////////ANALISIS DE SOLICITUD/////////////////////////////
if (strcasecmp(method, "GET") != 0)
send_error(io, "501", "Not supported", NULL, "Method is not supported.", path);
```

5.5.5. Respuesta

El siguiente paso es escribir una respuesta y creará la conexión. Nota que para esto usaremos **BIO_puts** y **BIO_write** . Esto nos permite escribir la respuesta una linea a la vez pero se obtiene la respuesta completa como si fuera un simple registro SSL. Esto es importante ya que el costo de preparar un registro SSL para transmisión resulta algo significativo.

```
void send_file(BIO *io, char *path, struct stat *statbuf)
{
char data[4096];
int n;
FILE *file = fopen(path, "r");
if (!file)
send_error(io, "403", "Forbidden", NULL, "Access denied.", path);
else
{
int length = S_ISREG(statbuf->st_mode) ? statbuf->st_size : -1;
send_headers(io, "200", "OK", NULL, get_mime_type(path), length, statbuf->st_mtime);
while ((n = fread(data, 1, sizeof(data), file)) > 0) BIO_write(io,data,n);
fclose(file);
}
}
```

5.5.6. Shutdown

Una vez que hemos terminado de transmitir la respuesta necesitamos enviar un *close_notify*. Esto es llevado a cabo usando **SSL_shutdown()**. De igual manera, cuando el cliente ha terminado y ha recibido el close_notify del servidor, el también tiene que mandar el suyo. Desafortunadamente, el cliente no siempre se comporta como se espera y puede mandar otro tipo de respuesta en lugar de close_notify. Es por eso que en nuestro código se prevé esta posibilidad y por eso se llama dos veces a **SSL_shutdown()**. Esto resuelve el problema y el cierre se puede lleva a cabo satisfactoriamente.

```
shutdown:
r=SSL_shutdown(ssl);
if(!r){
/* If we called SSL_shutdown() first then
we always get return value of '0'. In
this case, try again, but first send a
TCP FIN to trigger the other side's
close_notify*/
shutdown(s,1);
r=SSL_shutdown(ssl);
}
switch(r){
case 1:
break; /* Success */
case 0:
case -1:
default:
//berr_exit("Shutdown failed");
berr_exit("\n");
}
```

5.5.7. Compilación y ejecución

Así como lo hicimos en el *capitulo 3*, para compilar el kernel de Linux, haremos una compilación cruzada sobre el servidor. Para tener un mejor control, nuestro servidor se compone de varios archivos los cuales están interrelacionados entre si y estos son:

- common.h
- common.c
- server.h
- server.c
- wserver.c

Estos archivos los vamos a tener contenidos en el directorio $\sim$/proyecto/servidor que creamos con anterioridad junto con los archivos root.pem, server.pem y dh-param.pem en el *capitulo 4*. Aquí es donde realizaremos la compilación.

5.5.7.1. Dependencias

En primer lugar, para llevar a cabo la compilación cruzada de nuestro servidor, de nuevo haremos uso de nuestra toolchain pero para que esto funcione debemos agregarle algunas librerías de desarrollo necesarias para OpenSSL así como sus respectivas dependencias todas estas para la arquitectura Powerpc. Necesitaremos los siguientes paquetes:

- libssl
- libssl-dev
- xlib1g

Estas las podemos obtener desde la pagina https://packages.debian.org/. Aquí mostraré las instrucciones necesarias para descargarlas:

: ~/proyecto$ wget http://ftp.us.debian.org/debian/pool/main/z/zlib/zlib1g_1.2.3.4.df

3_powerpc.deb

: ~/proyecto$ wget http://ftp.us.debian.org/debian/pool/main/o/openssl/libssl0.9.8_0.

4squeeze14_powerpc.deb

: ~/proyecto$ wget http://ftp.us.debian.org/debian/pool/main/o/openssl/libssl-

dev_0.9.8o-4squeeze14_powerpc.deb

El programa predeterminado para instalar estos paquetes en un sistema Debian es **dpkg** pero nosotros tendremos que instalar los paquetes manualmente en nuestra toolchain, lo que significa copiar los archivos contenidos en cada paquete en los directorios adecuados para que la toolchain los pueda localizar al momento de llevar a cabo la compilación. En cada paquete vine establecido el directorio en donde debe ir contenido cada archivo. Ya habiendo especificado esto, procedemos a la instalación:

1. Extraemos paquetes:

Para zlib1g:

: ~/proyecto$ dpkg -x zlib1g_1.2.3.4.dfsg-3_powerpc.deb ./

: ~/proyecto$ cp usr/lib/* toolchain/powerpc-linux-gnu/libc/usr/lib

: ~/proyecto$ cp -r usr/share/doc toolchain/powerpc-linux-gnu/libc/usr/share

: ~/proyecto$ rm -r usr/

Para libssl:

: ~/proyecto$ dpkg -x libssl0.9.8_0.9.8o-4squeeze14_powerpc.deb

./

: ~/proyecto$ cp -r usr/lib/* toolchain/powerpc-linux-gnu/libc/usr/lib

: ~/proyecto$ cp -r usr/share/doc/libssl0.9.8 toolchain/powerpc-linux-gnu/libc/usr/share/doc

: ~/proyecto$ rm -r usr/

Para libssl-dev:

: ~/proyecto$ dpkg -x libssl-dev_0.9.8o-4squeeze14_powerpc.deb ./

: ~/proyecto$ cp -r usr/lib/* toolchain/powerpc-linux-gnu/libc/usr/lib

: ~/proyecto$ cp -r usr/include/* toolchain/powerpc-linux-gnu/libc/usr/include

: ~/proyecto$ cp -r usr/share/doc/libssl-dev toolchain/powerpc-linux-gnu/libc/usr/share/doc

: ~/proyecto$ cp -r usr/share/man/man3 toolchain/powerpc-linux-gnu/libc/usr/share/man

: ~/proyecto$ rm -r usr/

Es importante señalar que este mismo procedimiento debe llevarse a cabo sobre el sistema de archivos que se generó con Buildroot en el *capitulo 3* para que nuestro servidor pueda ejecutarse.

5.5.7.2. Compilación

Ahora cargamos las variables de ambiente para su utilización:

: ~/proyecto$ source $\sim$/proyecto/toolchain/ambiente.sh

Y ahora ya podemos compilar de la siguiente forma:

: ~/proyecto$ cd servidor/

: ~/proyecto/servidor$ powerpc-linux-gnu-gcc -Wall -pthread common.c server.c wserver.c -o server -lssl -lcrypto

Esto nos generará el ejecutable *server* y podemos revisar sus características con el siguiente comando:

: ~/proyecto/servidor$ readelf -h server

lo que nos producirá la siguiente salida:

```
ELF Header:
Magic: 7f 45 4c 46 01 02 01 00 00 00 00 00 00 00 00 00
Class: ELF32
```

```
Data: 2's complement, big endian
Version: 1 (current)
OS/ABI: UNIX - System V
ABI Version: 0
Type: EXEC (Executable file)
Machine: PowerPC
Version: 0x1
Entry point address: 0x10001240
Start of program headers: 52 (bytes into file)
Start of section headers: 14624 (bytes into file)
Flags: 0x0
Size of this header: 52 (bytes)
Size of program headers: 32 (bytes)
Number of program headers: 8
Size of section headers: 40 (bytes)
Number of section headers: 32
Section header string table index: 29
```

lo que mas nos interesa es la linea *Machine: PowerPC* que indica que el archivo se ejecuta sobre una arquitectura PowerPC. Este archivo, junto con *dhparam.pem, root.pem y server.pem*, los mandamos a nuestro sistema de archivos generado en el *capitulo 3* y desde allí lo ejecutamos.

Para esto primero tenemos que montar el sistema de archivos y posteriormente copiarlo a un directorio apropiado para ello:

: ∼/proyecto/servidor$ sudo mount /dev/DispositivoCompactFlash2 /mnt

: ∼/proyecto/servidor$ mkdir /mnt/root/servidor

: ∼/proyecto/servidor$ cp server dhparam.pem root.pem server.pem /mnt/root/servidor/

: ∼/proyecto/servidor$ umount /mnt

Ya habiendo hecho esto ejecutamos minicom y cargamos nuestro sistema en la tarjeta

: ∼/proyecto$ sudo minicom

Ya estando en el sistema ejecutamos el servidor:

: ∼# cd /root/servidor/

: ∼/root/servidor# ../server

Ahora podremos mandar solicitudes a nuestro servidor desde cualquier host conectado a nuestra red local por medio de un navegador web como por ejemplo firefox. Solo tenemos que ingresar la dirección web de nuestro servidor en la barra de direcciones del navegador que en nuestro caso es la **dirección ip** que obtuvo la tarjeta junto con el **puerto** 4433, antecedida por el protocolo que estamos utilizando: **https**.

El navegador nos mostrará un mensaje de advertencia indicando que la conexión no es confiable. Esto pasa ya que el servidor esta tratando de autenticarse a si mismo como vimos en el *capitulo 4*.

Figura 5.1: Inicio de conexión

Hacemos clic en "Entiendo los riesgos" y después en agregar excepción.

Figura 5.2: Agregar excepción de seguridad

Se nos abrirá una ventana para añadir la excepción de seguridad. Damos clic

en obtener certificado y opcionalmente en ver. Dejamos seleccionada la opción de Guardar excepción permanentemente y por último confirmamos la excepción.

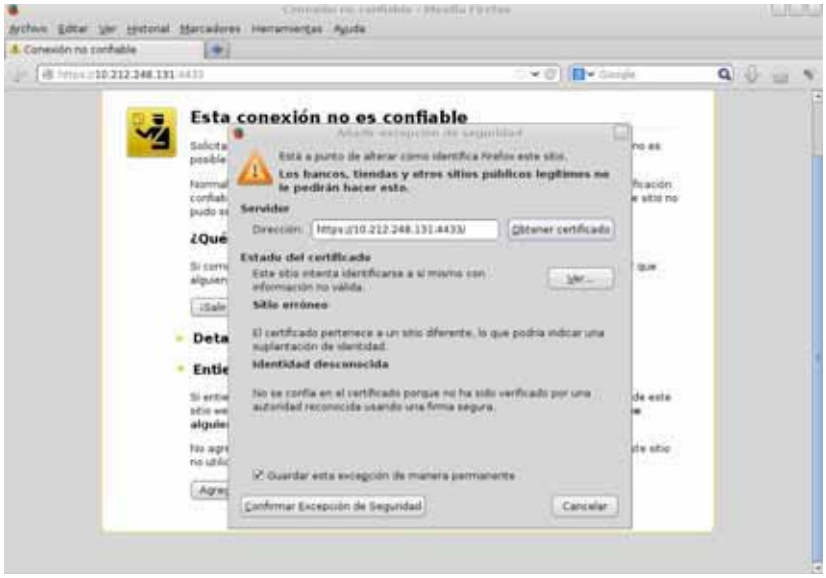

Figura 5.3: Confirmar excepción de seguridad

Figura 5.4: Certificado

Se nos mostrará la página de inicio del servidor "index.html". El diseño de esta página de inicio y los archivos a los que dirijan los links de esta, quedan fuera del objetivo de este proyecto, sin embargo, deben existir y alojarse en el directorio raíz de nuestro sistema de archivos para que el servidor funcione correctamente.

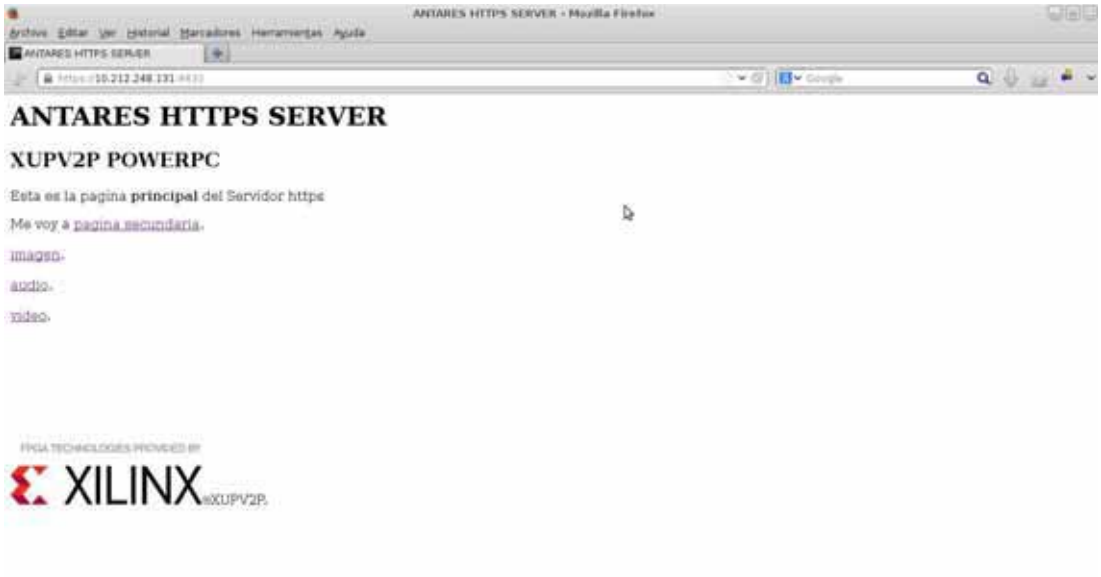

Figura 5.5: Página de inicio del servidor

Para verificar que nuestra conexión es segura, haremos clic en el icono de candado que esta en la parte izquierda de la barra de direcciones del navegador y damos clic en "más información...". Se nos mostrará toda la información acerca de nuestra conexión.

Figura 5.6: Detalles de conexión

Bibliografía y recursos

- "Embedded Systems Design With Platform FPGAS", Ronald R. Sass, Andrew G. Schmidt, Ed. Elsevier
- "Linux: Manual de referencia", Petersen, Richard, 6a Edición, Ed. Mc Graw Hill
- http://buildroot.uclibc.org/
- http://elinux.org/Main_Page
- "Comunicaciones y Redes de Computadores", William Stallings, 7a Edición, Ed. Pearson Prentice Hall
- "Network Security with OpenSSL", John Viega, Matt Messier, Pravir Chandra, Ed. O'Reilly
- "SSH The Secure Shell", Daniel J. Barrett, Richard E Silverman, 2a Edición, Ed. O'Reilly
- "SSL and TLS: Designing and Building Secure Systems", Eric Rescorla, Ed Addison-Wesley
- http://www.linuxjournal.com/
- Linux Journal Issue #89/September 2001
- https://www.openssl.org/
- http://www.madboa.com/geek/openssl/
- "UNIX Programación avanzada", Francisco M. Márquez, 3a Edición, Ed. Alfaomega,

Apéndice A

Instalación ISE y EDK

Instalación ISE Se abrirá la pantalla de inicio que nos informa acerca de la clave de registro que debemos tener para la instalación. Nosotros ya tenemos nuestra clave asi que continuamos a la siguiente pantalla donde aceptaremos los acuerdos de la licencia.

Figura A.1: Pantalla de inicio

Figura A.2: Acuerdo de licencia

Posteriormente nos pedirá la clave de registro y una vez ingresada nos pedirá el directorio donde deseamos instalar el software el cual será /opt, esto nos creara un directorio llamado Xilinx en la ruta especificada.

Figura A.3: Clave de registro

Nos pedirá seleccionar los módulos que necesitaremos, en nuestro caso son

solo 4.Esto nos permitirá trabajar con la tarjeta XUP Virtex II Pro y nos proporcionará herramientas de diseño

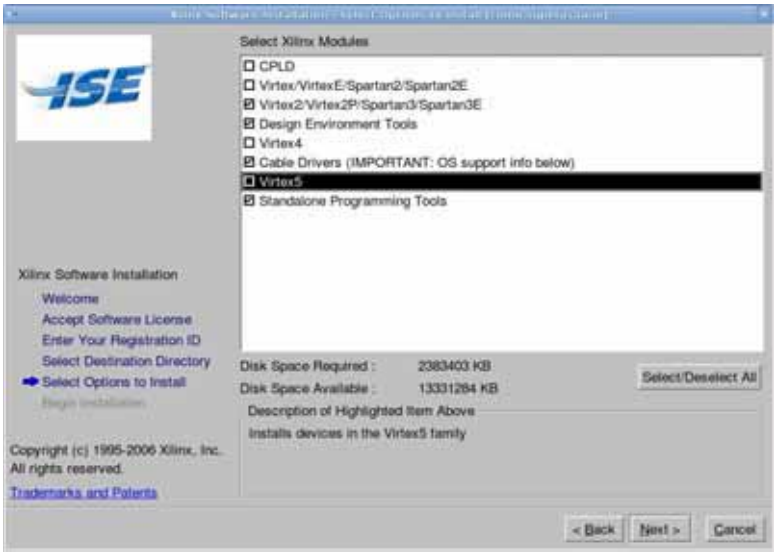

Figura A.4: Selección de módulos

Figura A.5: Ruta de instalación

Ahora seleccionaremos las variables de ambiente que utilizaremos. Dejaremos marcadas todas las casillas.

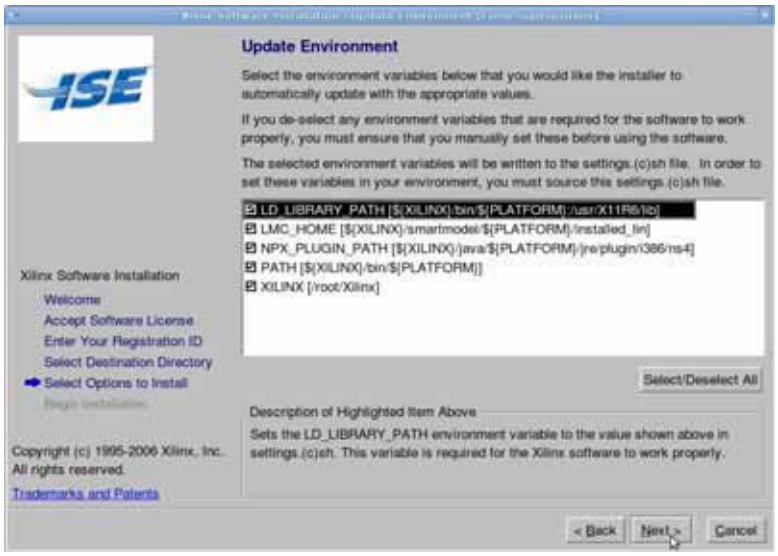

Figura A.6: Selección de variables de ambiente

Se nos mostrará un resumen de toda la configuración, revisamos que todo sea correcto y damos Instalar.

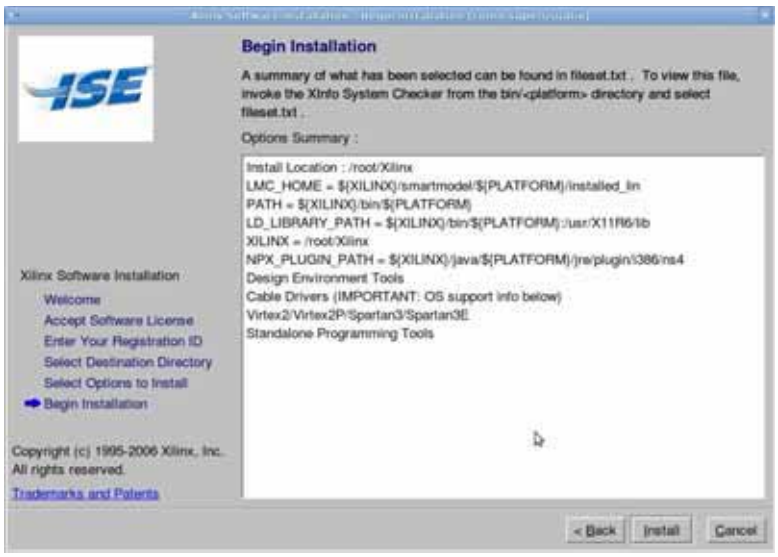

Figura A.7: Resumen de configuración

Instalación EDK Esta instalación es muy similar a la anterior, nos aparecerá la misma pantalla de inicio, nos pedirá aceptar el acuerdo de licencia e ingresar

la clave de registro.

Figura A.8: Pantalla de inicio

Figura A.9: Acuerdo de licencia

Figura A.10: Ruta de instalación

Figura A.11: Clave de registro

Se nos pedirá también que seleccionamos los módulos de Xilinx. Dejaremos seleccionado el único modulo que aparece.

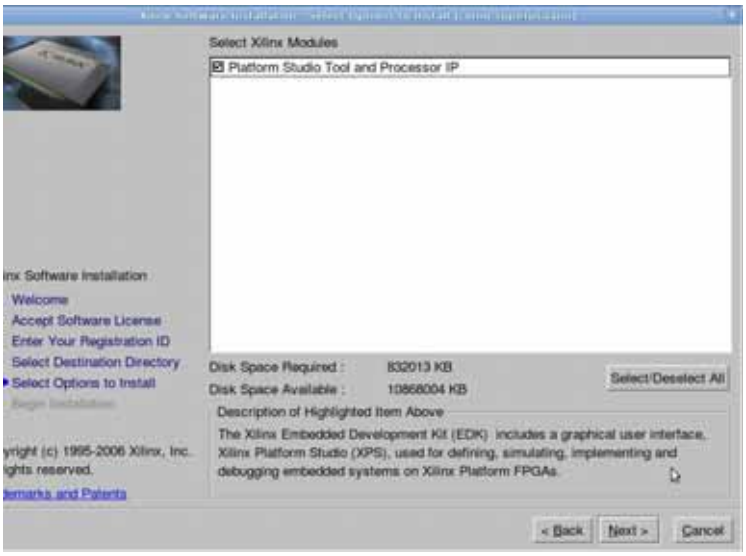

Figura A.12: Selección de modulos

Después dejaremos seleccionadas todas las variables de ambiente que se muestren. Estas son necesarias para poder lanzar la aplicación, tanto ISE como EDK.

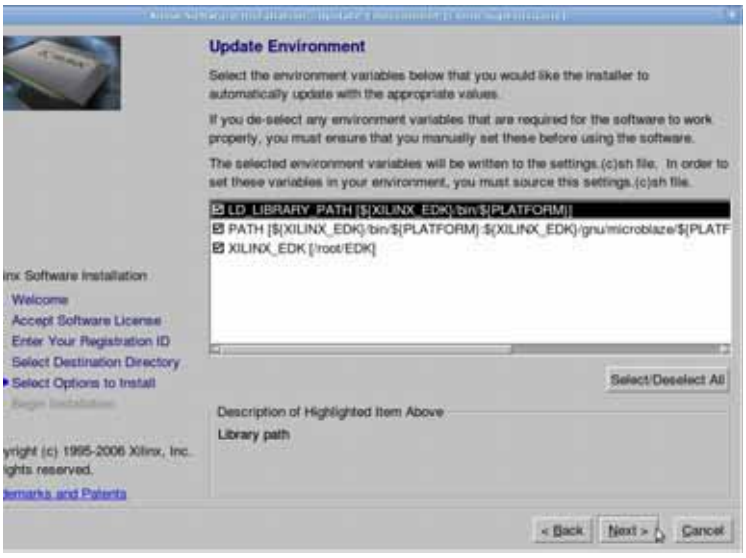

Figura A.13: Selección de variables de ambiente

Nos mostrará un resumen de todo lo seleccionado y le damos instalar si todo

esta bien.

Figura A.14: Resumen de instalación

Apéndice B

Códigos fuente

Configuración kernel (Archivo .config)

```
# Automatically generated file; DO NOT EDIT.
# Linux/powerpc 3.2.0 Kernel Configuration
#
# CONFIG_PPC64 is not set
#
# Processor support
#
# CONFIG_PPC_BOOK3S_32 is not set
# CONFIG_PPC_85xx is not set
# CONFIG_PPC_8xx is not set
CONFIG_40x=y
# CONFIG_44x is not set
# CONFIG_E200 is not set
CONFIG_4xx=y
CONFIG_PPC_MMU_NOHASH=y
# CONFIG_PPC_MM_SLICES is not set
CONFIG_NOT_COHERENT_CACHE=y
CONFIG_PPC32=y
CONFIG_32BIT=y
CONFIG_WORD_SIZE=32
# CONFIG_ARCH_PHYS_ADDR_T_64BIT is not set
# CONFIG_ARCH_DMA_ADDR_T_64BIT is not set
CONFIG_MMU=y
CONFIG_GENERIC_CMOS_UPDATE=y
CONFIG_GENERIC_TIME_VSYSCALL=y
CONFIG_GENERIC_CLOCKEVENTS=y
# CONFIG_HAVE_SETUP_PER_CPU_AREA is not set
# CONFIG_NEED_PER_CPU_EMBED_FIRST_CHUNK is not set
CONFIG_NR_IRQS=512
CONFIG_STACKTRACE_SUPPORT=y
CONFIG_HAVE_LATENCYTOP_SUPPORT=y
CONFIG_TRACE_IRQFLAGS_SUPPORT=y
CONFIG_LOCKDEP_SUPPORT=y
CONFIG_RWSEM_XCHGADD_ALGORITHM=y
CONFIG_ARCH_HAS_ILOG2_U32=y
CONFIG_GENERIC_HWEIGHT=y
CONFIG_GENERIC_GPIO=y
# CONFIG_ARCH_NO_VIRT_TO_BUS is not set
CONFIG_PPC=y
CONFIG_EARLY_PRINTK=y
CONFIG_GENERIC_NVRAM=y
CONFIG_SCHED_OMIT_FRAME_POINTER=y
CONFIG_ARCH_MAY_HAVE_PC_FDC=y
CONFIG_PPC_OF=y
CONFIG_PPC_UDBG_16550=y
# CONFIG_GENERIC_TBSYNC is not set
CONFIG_AUDIT_ARCH=y
CONFIG_GENERIC_BUG=y
# CONFIG_EPAPR_BOOT is not set
CONFIG_DEFAULT_UIMAGE=y
CONFIG_ARCH_HIBERNATION_POSSIBLE=y
CONFIG_ARCH_SUSPEND_POSSIBLE=y
CONFIG_PPC_DCR_NATIVE=y
# CONFIG_PPC_DCR_MMIO is not set
CONFIG_PPC_DCR=y
CONFIG_ARCH_SUPPORTS_DEBUG_PAGEALLOC=y
CONFIG_PPC_ADV_DEBUG_REGS=y
CONFIG_PPC_ADV_DEBUG_IACS=2
CONFIG_PPC_ADV_DEBUG_DACS=2
CONFIG_PPC_ADV_DEBUG_DVCS=0
```

```
CONFIG_DEFCONFIG_LIST="/lib/modules/$UNAME_RELEASE/.config"
CONFIG_HAVE_IRQ_WORK=y
CONFIG_IRQ_WORK=y
#
# General setup
#
CONFIG_EXPERIMENTAL=y
CONFIG_BROKEN_ON_SMP=y
CONFIG_INIT_ENV_ARG_LIMIT=32
CONFIG_CROSS_COMPILE=""
CONFIG_LOCALVERSION=""
# CONFIG_LOCALVERSION_AUTO is not set
CONFIG_DEFAULT_HOSTNAME="(none)"
# CONFIG_SWAP is not set
CONFIG_SYSVIPC=y
CONFIG_SYSVIPC_SYSCTL=y
CONFIG_POSIX_MQUEUE=y
CONFIG_POSIX_MQUEUE_SYSCTL=y
CONFIG_BSD_PROCESS_ACCT=y
CONFIG_BSD_PROCESS_ACCT_V3=y
# CONFIG_FHANDLE is not set
CONFIG_TASKSTATS=y
CONFIG_TASK_DELAY_ACCT=y
CONFIG_TASK_XACCT=y
CONFIG_TASK_IO_ACCOUNTING=y
CONFIG_AUDIT=y
CONFIG_AUDITSYSCALL=y
CONFIG_AUDIT_WATCH=y
CONFIG_AUDIT_TREE=y
CONFIG_HAVE_GENERIC_HARDIRQS=y
#
# IRQ subsystem
#
CONFIG_GENERIC_HARDIRQS=y
CONFIG_HAVE_SPARSE_IRQ=y
CONFIG_GENERIC_IRQ_SHOW=y
CONFIG_GENERIC_IRQ_SHOW_LEVEL=y
CONFIG_SPARSE_IRQ=y
#
# RCU Subsystem
#
CONFIG_TINY_RCU=y
# CONFIG_PREEMPT_RCU is not set
# CONFIG_RCU_TRACE is not set
# CONFIG_TREE_RCU_TRACE is not set
# CONFIG_IKCONFIG is not set
CONFIG_LOG_BUF_SHIFT=17
CONFIG_CGROUPS=y
# CONFIG_CGROUP_DEBUG is not set
CONFIG_CGROUP_FREEZER=y
CONFIG_CGROUP_DEVICE=y
CONFIG_CPUSETS=y
CONFIG_PROC_PID_CPUSET=y
CONFIG_CGROUP_CPUACCT=y
# CONFIG_RESOURCE_COUNTERS is not set
# CONFIG_CGROUP_PERF is not set
CONFIG_CGROUP_SCHED=y
CONFIG_FAIR_GROUP_SCHED=y
# CONFIG_CFS_BANDWIDTH is not set
# CONFIG_RT_GROUP_SCHED is not set
# CONFIG_BLK_CGROUP is not set
CONFIG_NAMESPACES=y
CONFIG_UTS_NS=y
CONFIG_IPC_NS=y
CONFIG_USER_NS=y
CONFIG_PID_NS=y
CONFIG_NET_NS=y
# CONFIG_SCHED_AUTOGROUP is not set
# CONFIG_SYSFS_DEPRECATED is not set
CONFIG_RELAY=y
CONFIG_BLK_DEV_INITRD=y
CONFIG_INITRAMFS_SOURCE=""
CONFIG_RD_GZIP=y
# CONFIG_RD_BZIP2 is not set
# CONFIG_RD_LZMA is not set
# CONFIG_RD_XZ is not set
# CONFIG_RD_LZO is not set
CONFIG_CC_OPTIMIZE_FOR_SIZE=y
CONFIG_SYSCTL=y
CONFIG_ANON_INODES=y
CONFIG_EXPERT=y
# CONFIG_SYSCTL_SYSCALL is not set
CONFIG_KALLSYMS=y
# CONFIG_KALLSYMS_ALL is not set
CONFIG_HOTPLUG=y
CONFIG_PRINTK=y
CONFIG_BUG=y
CONFIG_ELF_CORE=y
CONFIG_BASE_FULL=y
CONFIG_FUTEX=y
CONFIG_EPOLL=y
```

```
CONFIG_SIGNALFD=y
CONFIG_TIMERFD=y
CONFIG_EVENTFD=y
CONFIG_SHMEM=y
CONFIG_AIO=y
CONFIG_EMBEDDED=y
CONFIG_HAVE_PERF_EVENTS=y
#
# Kernel Performance Events And Counters
#
CONFIG_PERF_EVENTS=y
# CONFIG_PERF_COUNTERS is not set
# CONFIG_DEBUG_PERF_USE_VMALLOC is not set
CONFIG_VM_EVENT_COUNTERS=y
CONFIG_SLUB_DEBUG=y
# CONFIG_COMPAT_BRK is not set
# CONFIG_SLAB is not set
CONFIG_SLUB=y
# CONFIG_SLOB is not set
CONFIG_PROFILING=y
CONFIG_TRACEPOINTS=y
CONFIG_OPROFILE=m
CONFIG_HAVE_OPROFILE=y
CONFIG_KPROBES=y
# CONFIG_JUMP_LABEL is not set
CONFIG_HAVE_EFFICIENT_UNALIGNED_ACCESS=y
CONFIG_KRETPROBES=y
CONFIG_HAVE_IOREMAP_PROT=y
CONFIG_HAVE_KPROBES=y
CONFIG_HAVE_KRETPROBES=y
CONFIG_HAVE_ARCH_TRACEHOOK=y
CONFIG_HAVE_DMA_ATTRS=y
CONFIG_HAVE_REGS_AND_STACK_ACCESS_API=y
CONFIG_HAVE_DMA_API_DEBUG=y
CONFIG_HAVE_ARCH_JUMP_LABEL=y
CONFIG_ARCH_HAVE_NMI_SAFE_CMPXCHG=y
#
# GCOV-based kernel profiling
#
# CONFIG_GCOV_KERNEL is not set
# CONFIG_HAVE_GENERIC_DMA_COHERENT is not set
CONFIG_SLABINFO=y
CONFIG_RT_MUTEXES=y
CONFIG_BASE_SMALL=0
CONFIG_MODULES=y
CONFIG_MODULE_FORCE_LOAD=y
CONFIG_MODULE_UNLOAD=y
CONFIG_MODULE_FORCE_UNLOAD=y
CONFIG_MODVERSIONS=y
# CONFIG_MODULE_SRCVERSION_ALL is not set
CONFIG_BLOCK=y
CONFIG_LBDAF=y
CONFIG_BLK_DEV_BSG=y
CONFIG_BLK_DEV_BSGLIB=y
CONFIG_BLK_DEV_INTEGRITY=y
#
# IO Schedulers
#
CONFIG_IOSCHED_NOOP=y
CONFIG_IOSCHED_DEADLINE=y
CONFIG_IOSCHED_CFQ=y
# CONFIG_DEFAULT_DEADLINE is not set
CONFIG_DEFAULT_CFQ=y
# CONFIG_DEFAULT_NOOP is not set
CONFIG_DEFAULT_IOSCHED="cfq"
# CONFIG_INLINE_SPIN_TRYLOCK is not set
# CONFIG_INLINE_SPIN_TRYLOCK_BH is not set
# CONFIG_INLINE_SPIN_LOCK is not set
# CONFIG_INLINE_SPIN_LOCK_BH is not set
# CONFIG_INLINE_SPIN_LOCK_IRQ is not set
# CONFIG_INLINE_SPIN_LOCK_IRQSAVE is not set
CONFIG_INLINE_SPIN_UNLOCK=y
# CONFIG_INLINE_SPIN_UNLOCK_BH is not set
CONFIG_INLINE_SPIN_UNLOCK_IRQ=y
# CONFIG_INLINE_SPIN_UNLOCK_IRQRESTORE is not set
# CONFIG_INLINE_READ_TRYLOCK is not set
# CONFIG_INLINE_READ_LOCK is not set
# CONFIG_INLINE_READ_LOCK_BH is not set
# CONFIG_INLINE_READ_LOCK_IRQ is not set
# CONFIG_INLINE_READ_LOCK_IRQSAVE is not set
CONFIG_INLINE_READ_UNLOCK=y
# CONFIG_INLINE_READ_UNLOCK_BH is not set
CONFIG_INLINE_READ_UNLOCK_IRQ=y
# CONFIG_INLINE_READ_UNLOCK_IRQRESTORE is not set
# CONFIG_INLINE_WRITE_TRYLOCK is not set
# CONFIG_INLINE_WRITE_LOCK is not set
# CONFIG_INLINE_WRITE_LOCK_BH is not set
# CONFIG_INLINE_WRITE_LOCK_IRQ is not set
# CONFIG_INLINE_WRITE_LOCK_IRQSAVE is not set
CONFIG_INLINE_WRITE_UNLOCK=y
# CONFIG_INLINE_WRITE_UNLOCK_BH is not set
```

```
CONFIG_INLINE_WRITE_UNLOCK_IRQ=y
# CONFIG_INLINE_WRITE_UNLOCK_IRQRESTORE is not set
# CONFIG_MUTEX_SPIN_ON_OWNER is not set
CONFIG_FREEZER=y
# CONFIG_PPC_XICS is not set
# CONFIG_PPC_ICP_NATIVE is not set
# CONFIG_PPC_ICP_HV is not set
# CONFIG_PPC_ICS_RTAS is not set
#
# Platform support
#
# CONFIG_PPC_CELL is not set
# CONFIG_PPC_CELL_NATIVE is not set
# CONFIG_PQ2ADS is not set
# CONFIG_ISS4xx is not set
# CONFIG_PPC4xx_GPIO is not set
CONFIG_XILINX_VIRTEX=y
# CONFIG_ACADIA is not set
# CONFIG_EP405 is not set
# CONFIG_HOTFOOT is not set
# CONFIG_KILAUEA is not set
# CONFIG_MAKALU is not set
# CONFIG_WALNUT is not set
CONFIG_XILINX_VIRTEX_GENERIC_BOARD=y
# CONFIG_PPC40x_SIMPLE is not set
CONFIG_XILINX_VIRTEX_II_PRO=y
CONFIG_XILINX_VIRTEX_4_FX=y
CONFIG_IBM405_ERR77=y
CONFIG_IBM405_ERR51=y
# CONFIG_PPC_WSP is not set
# CONFIG_KVM_GUEST is not set
# CONFIG_IPIC is not set
# CONFIG_MPIC is not set
# CONFIG_PPC_EPAPR_HV_PIC is not set
# CONFIG_MPIC_WEIRD is not set
# CONFIG_PPC_I8259 is not set
# CONFIG_PPC_RTAS is not set
# CONFIG_MMIO_NVRAM is not set
# CONFIG_MPIC_U3_HT_IRQS is not set
# CONFIG_PPC_MPC106 is not set
# CONFIG_PPC_970_NAP is not set
# CONFIG_PPC_P7_NAP is not set
#
# CPU Frequency scaling
#
CONFIG_CPU_FREQ=y
CONFIG_CPU_FREQ_TABLE=y
CONFIG_CPU_FREQ_STAT=m
# CONFIG_CPU_FREQ_STAT_DETAILS is not set
# CONFIG_CPU_FREQ_DEFAULT_GOV_PERFORMANCE is not set
# CONFIG_CPU_FREQ_DEFAULT_GOV_POWERSAVE is not set
# CONFIG_CPU_FREQ_DEFAULT_GOV_USERSPACE is not set
CONFIG_CPU_FREQ_DEFAULT_GOV_ONDEMAND=y
# CONFIG_CPU_FREQ_DEFAULT_GOV_CONSERVATIVE is not set
CONFIG_CPU_FREQ_GOV_PERFORMANCE=y
CONFIG_CPU_FREQ_GOV_POWERSAVE=m
CONFIG_CPU_FREQ_GOV_USERSPACE=m
CONFIG_CPU_FREQ_GOV_ONDEMAND=y
CONFIG_CPU_FREQ_GOV_CONSERVATIVE=m
#
# PowerPC CPU frequency scaling drivers
#
#
# CPU Frequency drivers
#
# CONFIG_FSL_ULI1575 is not set
# CONFIG_SIMPLE_GPIO is not set
#
# Kernel options
#
# CONFIG_HIGHMEM is not set
CONFIG_TICK_ONESHOT=y
CONFIG_NO_HZ=y
CONFIG_HIGH_RES_TIMERS=y
CONFIG_GENERIC_CLOCKEVENTS_BUILD=y
# CONFIG_HZ_100 is not set
CONFIG_HZ_250=y
# CONFIG_HZ_300 is not set
# CONFIG_HZ_1000 is not set
CONFIG_HZ=250
CONFIG_SCHED_HRTICK=y
# CONFIG_PREEMPT_NONE is not set
CONFIG_PREEMPT_VOLUNTARY=y
# CONFIG_PREEMPT is not set
CONFIG_BINFMT_ELF=y
CONFIG_CORE_DUMP_DEFAULT_ELF_HEADERS=y
# CONFIG_HAVE_AOUT is not set
CONFIG_BINFMT_MISC=m
CONFIG_MATH_EMULATION=y
CONFIG_IOMMU_HELPER=y
CONFIG_SWIOTLB=y
```

```
CONFIG_ARCH_ENABLE_MEMORY_HOTPLUG=y
CONFIG_ARCH_HAS_WALK_MEMORY=y
CONFIG_ARCH_ENABLE_MEMORY_HOTREMOVE=y
CONFIG_MAX_ACTIVE_REGIONS=32
CONFIG_ARCH_FLATMEM_ENABLE=y
CONFIG_ARCH_POPULATES_NODE_MAP=y
CONFIG_SELECT_MEMORY_MODEL=y
CONFIG_FLATMEM_MANUAL=y
CONFIG_FLATMEM=y
CONFIG_FLAT_NODE_MEM_MAP=y
CONFIG_HAVE_MEMBLOCK=y
CONFIG_PAGEFLAGS_EXTENDED=y
CONFIG_SPLIT_PTLOCK_CPUS=4
# CONFIG_COMPACTION is not set
CONFIG_MIGRATION=y
# CONFIG_PHYS_ADDR_T_64BIT is not set
CONFIG_ZONE_DMA_FLAG=1
CONFIG_BOUNCE=y
CONFIG_VIRT_TO_BUS=y
CONFIG_KSM=y
CONFIG_DEFAULT_MMAP_MIN_ADDR=65536
CONFIG_NEED_PER_CPU_KM=y
# CONFIG_CLEANCACHE is not set
CONFIG_PPC_4K_PAGES=y
CONFIG_FORCE_MAX_ZONEORDER=11
CONFIG_CMDLINE_BOOL=y
CONFIG_CMDLINE="console=ttyS0,9600 console=tty0 root=/dev/sda2"
CONFIG_EXTRA_TARGETS="simpleImage.virtex405-antares simpleImage.initrd.virtex405-antares"
CONFIG_SUSPEND=y
CONFIG_SUSPEND_FREEZER=y
CONFIG_PM_SLEEP=y
CONFIG_PM_RUNTIME=y
CONFIG_PM=y
CONFIG_PM_DEBUG=y
# CONFIG_PM_ADVANCED_DEBUG is not set
# CONFIG_PM_TEST_SUSPEND is not set
CONFIG_CAN_PM_TRACE=y
CONFIG_SECCOMP=y
CONFIG_ISA_DMA_API=y
#
# Bus options
#
CONFIG_ZONE_DMA=y
CONFIG_NEED_DMA_MAP_STATE=y
CONFIG_NEED_SG_DMA_LENGTH=y
CONFIG_GENERIC_ISA_DMA=y
CONFIG_PPC4xx_CPM=y
CONFIG_4xx_SOC=y
CONFIG_PPC_PCI_CHOICE=y
# CONFIG_PCI is not set
# CONFIG_PCI_DOMAINS is not set
# CONFIG_PCI_SYSCALL is not set
# CONFIG_ARCH_SUPPORTS_MSI is not set
# CONFIG_PCCARD is not set
# CONFIG_HAS_RAPIDIO is not set
#
# Advanced setup
#
# CONFIG_ADVANCED_OPTIONS is not set
#
# Default settings for advanced configuration options are used
#
CONFIG_LOWMEM_SIZE=0x30000000
CONFIG_PAGE_OFFSET=0xc0000000
CONFIG_KERNEL_START=0xc0000000
CONFIG_PHYSICAL_START=0x00000000
CONFIG_TASK_SIZE=0xc0000000
CONFIG_CONSISTENT_SIZE=0x00200000
CONFIG_NET=y
#
# Networking options
#
CONFIG_PACKET=y
CONFIG_UNIX=y
CONFIG_XFRM=y
CONFIG_XFRM_USER=m
CONFIG_XFRM_SUB_POLICY=y
CONFIG_XFRM_MIGRATE=y
# CONFIG_XFRM_STATISTICS is not set
CONFIG_XFRM_IPCOMP=m
CONFIG_NET_KEY=m
CONFIG_NET_KEY_MIGRATE=y
CONFIG_INET=y
CONFIG_IP_MULTICAST=y
CONFIG_IP_ADVANCED_ROUTER=y
# CONFIG_IP_FIB_TRIE_STATS is not set
CONFIG_IP_MULTIPLE_TABLES=y
CONFIG_IP_ROUTE_MULTIPATH=y
CONFIG_IP_ROUTE_VERBOSE=y
CONFIG_IP_ROUTE_CLASSID=y
CONFIG_IP_PNP=y
```

```
CONFIG_IP_PNP_DHCP=y
CONFIG_IP_PNP_BOOTP=y
CONFIG_IP_PNP_RARP=y
CONFIG_NET_IPIP=m
# CONFIG_NET_IPGRE_DEMUX is not set
CONFIG_IP_MROUTE=y
# CONFIG_IP_MROUTE_MULTIPLE_TABLES is not set
CONFIG_IP_PIMSM_V1=y
CONFIG_IP_PIMSM_V2=y
# CONFIG_ARPD is not set
CONFIG_SYN_COOKIES=y
CONFIG_INET_AH=m
CONFIG_INET_ESP=m
CONFIG_INET_IPCOMP=m
CONFIG_INET_XFRM_TUNNEL=m
CONFIG_INET_TUNNEL=m
CONFIG_INET_XFRM_MODE_TRANSPORT=m
CONFIG_INET_XFRM_MODE_TUNNEL=m
CONFIG_INET_XFRM_MODE_BEET=m
CONFIG_INET_LRO=m
CONFIG_INET_DIAG=m
CONFIG_INET_TCP_DIAG=m
CONFIG_TCP_CONG_ADVANCED=y
CONFIG_TCP_CONG_BIC=m
CONFIG_TCP_CONG_CUBIC=y
CONFIG_TCP_CONG_WESTWOOD=m
CONFIG_TCP_CONG_HTCP=m
CONFIG_TCP_CONG_HSTCP=m
CONFIG_TCP_CONG_HYBLA=m
CONFIG_TCP_CONG_VEGAS=m
CONFIG_TCP_CONG_SCALABLE=m
CONFIG_TCP_CONG_LP=m
CONFIG_TCP_CONG_VENO=m
CONFIG_TCP_CONG_YEAH=m
CONFIG_TCP_CONG_ILLINOIS=m
CONFIG_DEFAULT_CUBIC=y
# CONFIG_DEFAULT_RENO is not set
CONFIG_DEFAULT_TCP_CONG="cubic"
CONFIG_TCP_MD5SIG=y
CONFIG_IPV6=y
CONFIG_IPV6_PRIVACY=y
CONFIG_IPV6_ROUTER_PREF=y
CONFIG_IPV6_ROUTE_INFO=y
CONFIG_IPV6_OPTIMISTIC_DAD=y
CONFIG_INET6_AH=m
CONFIG_INET6_ESP=m
CONFIG_INET6_IPCOMP=m
CONFIG_IPV6_MIP6=y
CONFIG_INET6_XFRM_TUNNEL=m
CONFIG_INET6_TUNNEL=m
CONFIG_INET6_XFRM_MODE_TRANSPORT=m
CONFIG_INET6_XFRM_MODE_TUNNEL=m
CONFIG_INET6_XFRM_MODE_BEET=m
CONFIG_INET6_XFRM_MODE_ROUTEOPTIMIZATION=m
CONFIG_IPV6_SIT=m
# CONFIG_IPV6_SIT_6RD is not set
CONFIG_IPV6_NDISC_NODETYPE=y
CONFIG_IPV6_TUNNEL=m
CONFIG_IPV6_MULTIPLE_TABLES=y
CONFIG_IPV6_SUBTREES=y
CONFIG_IPV6_MROUTE=y
# CONFIG_IPV6_MROUTE_MULTIPLE_TABLES is not set
CONFIG_IPV6_PIMSM_V2=y
# CONFIG_NETLABEL is not set
CONFIG_NETWORK_SECMARK=y
# CONFIG_NETWORK_PHY_TIMESTAMPING is not set
CONFIG_NETFILTER=y
# CONFIG_NETFILTER_DEBUG is not set
CONFIG_NETFILTER_ADVANCED=y
CONFIG_BRIDGE_NETFILTER=y
#
# Core Netfilter Configuration
#
CONFIG_NETFILTER_NETLINK=m
CONFIG_NETFILTER_NETLINK_QUEUE=m
CONFIG_NETFILTER_NETLINK_LOG=m
CONFIG_NF_CONNTRACK=m
CONFIG_NF_CONNTRACK_MARK=y
CONFIG_NF_CONNTRACK_SECMARK=y
CONFIG_NF_CONNTRACK_EVENTS=y
# CONFIG_NF_CONNTRACK_TIMESTAMP is not set
CONFIG_NF_CT_PROTO_DCCP=m
CONFIG_NF_CT_PROTO_GRE=m
CONFIG_NF_CT_PROTO_SCTP=m
CONFIG_NF_CT_PROTO_UDPLITE=m
CONFIG_NF_CONNTRACK_AMANDA=m
CONFIG_NF_CONNTRACK_FTP=m
CONFIG_NF_CONNTRACK_H323=m
CONFIG_NF_CONNTRACK_IRC=m
CONFIG_NF_CONNTRACK_BROADCAST=m
CONFIG_NF_CONNTRACK_NETBIOS_NS=m
```

```
# CONFIG_NF_CONNTRACK_SNMP is not set
CONFIG_NF_CONNTRACK_PPTP=m
CONFIG_NF_CONNTRACK_SANE=m
CONFIG_NF_CONNTRACK_SIP=m
CONFIG_NF_CONNTRACK_TFTP=m
CONFIG_NF_CT_NETLINK=m
CONFIG_NETFILTER_TPROXY=m
CONFIG_NETFILTER_XTABLES=m
#
# Xtables combined modules
#
CONFIG_NETFILTER_XT_MARK=m
CONFIG_NETFILTER_XT_CONNMARK=m
#
# Xtables targets
#
# CONFIG_NETFILTER_XT_TARGET_AUDIT is not set
# CONFIG_NETFILTER_XT_TARGET_CHECKSUM is not set
CONFIG_NETFILTER_XT_TARGET_CLASSIFY=m
CONFIG_NETFILTER_XT_TARGET_CONNMARK=m
CONFIG_NETFILTER_XT_TARGET_CONNSECMARK=m
# CONFIG_NETFILTER_XT_TARGET_CT is not set
CONFIG_NETFILTER_XT_TARGET_DSCP=m
CONFIG_NETFILTER_XT_TARGET_HL=m
# CONFIG_NETFILTER_XT_TARGET_IDLETIMER is not set
CONFIG_NETFILTER_XT_TARGET_MARK=m
CONFIG_NETFILTER_XT_TARGET_NFLOG=m
CONFIG_NETFILTER_XT_TARGET_NFQUEUE=m
CONFIG_NETFILTER_XT_TARGET_NOTRACK=m
CONFIG_NETFILTER_XT_TARGET_RATEEST=m
# CONFIG_NETFILTER_XT_TARGET_TEE is not set
CONFIG_NETFILTER_XT_TARGET_TPROXY=m
CONFIG_NETFILTER_XT_TARGET_TRACE=m
CONFIG_NETFILTER_XT_TARGET_SECMARK=m
CONFIG_NETFILTER_XT_TARGET_TCPMSS=m
CONFIG_NETFILTER_XT_TARGET_TCPOPTSTRIP=m
#
# Xtables matches
#
# CONFIG_NETFILTER_XT_MATCH_ADDRTYPE is not set
CONFIG_NETFILTER_XT_MATCH_CLUSTER=m
CONFIG_NETFILTER_XT_MATCH_COMMENT=m
CONFIG_NETFILTER_XT_MATCH_CONNBYTES=m
CONFIG_NETFILTER_XT_MATCH_CONNLIMIT=m
CONFIG_NETFILTER_XT_MATCH_CONNMARK=m
CONFIG_NETFILTER_XT_MATCH_CONNTRACK=m
# CONFIG_NETFILTER_XT_MATCH_CPU is not set
CONFIG_NETFILTER_XT_MATCH_DCCP=m
# CONFIG_NETFILTER_XT_MATCH_DEVGROUP is not set
CONFIG_NETFILTER_XT_MATCH_DSCP=m
CONFIG_NETFILTER_XT_MATCH_ESP=m
CONFIG_NETFILTER_XT_MATCH_HASHLIMIT=m
CONFIG_NETFILTER_XT_MATCH_HELPER=m
CONFIG_NETFILTER_XT_MATCH_HL=m
CONFIG_NETFILTER_XT_MATCH_IPRANGE=m
# CONFIG_NETFILTER_XT_MATCH_IPVS is not set
CONFIG_NETFILTER_XT_MATCH_LENGTH=m
CONFIG_NETFILTER_XT_MATCH_LIMIT=m
CONFIG_NETFILTER_XT_MATCH_MAC=m
CONFIG_NETFILTER_XT_MATCH_MARK=m
CONFIG_NETFILTER_XT_MATCH_MULTIPORT=m
CONFIG_NETFILTER_XT_MATCH_OSF=m
CONFIG_NETFILTER_XT_MATCH_OWNER=m
CONFIG_NETFILTER_XT_MATCH_POLICY=m
CONFIG_NETFILTER_XT_MATCH_PHYSDEV=m
CONFIG_NETFILTER_XT_MATCH_PKTTYPE=m
CONFIG_NETFILTER_XT_MATCH_QUOTA=m
CONFIG_NETFILTER_XT_MATCH_RATEEST=m
CONFIG_NETFILTER_XT_MATCH_REALM=m
CONFIG_NETFILTER_XT_MATCH_RECENT=m
CONFIG_NETFILTER_XT_MATCH_SCTP=m
CONFIG_NETFILTER_XT_MATCH_SOCKET=m
CONFIG_NETFILTER_XT_MATCH_STATE=m
CONFIG_NETFILTER_XT_MATCH_STATISTIC=m
CONFIG_NETFILTER_XT_MATCH_STRING=m
CONFIG_NETFILTER_XT_MATCH_TCPMSS=m
CONFIG_NETFILTER_XT_MATCH_TIME=m
CONFIG_NETFILTER_XT_MATCH_U32=m
# CONFIG_IP_SET is not set
CONFIG_IP_VS=m
CONFIG_IP_VS_IPV6=y
# CONFIG_IP_VS_DEBUG is not set
CONFIG_IP_VS_TAB_BITS=12
#
# IPVS transport protocol load balancing support
#
CONFIG_IP_VS_PROTO_TCP=y
CONFIG_IP_VS_PROTO_UDP=y
CONFIG_IP_VS_PROTO_AH_ESP=y
CONFIG_IP_VS_PROTO_ESP=y
CONFIG_IP_VS_PROTO_AH=y
```

```
# CONFIG_IP_VS_PROTO_SCTP is not set
#
# IPVS scheduler
#
CONFIG_IP_VS_RR=m
CONFIG_IP_VS_WRR=m
CONFIG_IP_VS_LC=m
CONFIG_IP_VS_WLC=m
CONFIG_IP_VS_LBLC=m
CONFIG_IP_VS_LBLCR=m
CONFIG_IP_VS_DH=m
CONFIG_IP_VS_SH=m
CONFIG_IP_VS_SED=m
CONFIG_IP_VS_NQ=m
#
# IPVS application helper
#
CONFIG_IP_VS_FTP=m
CONFIG_IP_VS_NFCT=y
# CONFIG_IP_VS_PE_SIP is not set
#
# IP: Netfilter Configuration
#
CONFIG_NF_DEFRAG_IPV4=m
CONFIG_NF_CONNTRACK_IPV4=m
CONFIG_NF_CONNTRACK_PROC_COMPAT=y
CONFIG_IP_NF_QUEUE=m
CONFIG_IP_NF_IPTABLES=m
CONFIG_IP_NF_MATCH_AH=m
CONFIG_IP_NF_MATCH_ECN=m
CONFIG_IP_NF_MATCH_TTL=m
CONFIG_IP_NF_FILTER=m
CONFIG_IP_NF_TARGET_REJECT=m
CONFIG_IP_NF_TARGET_LOG=m
CONFIG_IP_NF_TARGET_ULOG=m
CONFIG_NF_NAT=m
CONFIG_NF_NAT_NEEDED=y
CONFIG_IP_NF_TARGET_MASQUERADE=m
CONFIG_IP_NF_TARGET_NETMAP=m
CONFIG_IP_NF_TARGET_REDIRECT=m
CONFIG_NF_NAT_PROTO_DCCP=m
CONFIG_NF_NAT_PROTO_GRE=m
CONFIG_NF_NAT_PROTO_UDPLITE=m
CONFIG_NF_NAT_PROTO_SCTP=m
CONFIG_NF_NAT_FTP=m
CONFIG_NF_NAT_IRC=m
CONFIG_NF_NAT_TFTP=m
CONFIG_NF_NAT_AMANDA=m
CONFIG_NF_NAT_PPTP=m
CONFIG_NF_NAT_H323=m
CONFIG_NF_NAT_SIP=m
CONFIG_IP_NF_MANGLE=m
CONFIG_IP_NF_TARGET_CLUSTERIP=m
CONFIG_IP_NF_TARGET_ECN=m
CONFIG_IP_NF_TARGET_TTL=m
CONFIG_IP_NF_RAW=m
CONFIG_IP_NF_SECURITY=m
CONFIG_IP_NF_ARPTABLES=m
CONFIG_IP_NF_ARPFILTER=m
CONFIG_IP_NF_ARP_MANGLE=m
#
# IPv6: Netfilter Configuration
#
CONFIG_NF_DEFRAG_IPV6=m
CONFIG_NF_CONNTRACK_IPV6=m
CONFIG_IP6_NF_QUEUE=m
CONFIG_IP6_NF_IPTABLES=m
CONFIG_IP6_NF_MATCH_AH=m
CONFIG_IP6_NF_MATCH_EUI64=m
CONFIG_IP6_NF_MATCH_FRAG=m
CONFIG_IP6_NF_MATCH_OPTS=m
CONFIG_IP6_NF_MATCH_HL=m
CONFIG_IP6_NF_MATCH_IPV6HEADER=m
CONFIG_IP6_NF_MATCH_MH=m
CONFIG_IP6_NF_MATCH_RT=m
CONFIG_IP6_NF_TARGET_HL=m
CONFIG_IP6_NF_TARGET_LOG=m
CONFIG_IP6_NF_FILTER=m
CONFIG_IP6_NF_TARGET_REJECT=m
CONFIG_IP6_NF_MANGLE=m
CONFIG_IP6_NF_RAW=m
CONFIG_IP6_NF_SECURITY=m
#
# DECnet: Netfilter Configuration
#
CONFIG_DECNET_NF_GRABULATOR=m
CONFIG_BRIDGE_NF_EBTABLES=m
CONFIG_BRIDGE_EBT_BROUTE=m
CONFIG_BRIDGE_EBT_T_FILTER=m
CONFIG_BRIDGE_EBT_T_NAT=m
CONFIG_BRIDGE_EBT_802_3=m
```

```
CONFIG_BRIDGE_EBT_AMONG=m
CONFIG_BRIDGE_EBT_ARP=m
CONFIG_BRIDGE_EBT_IP=m
CONFIG_BRIDGE_EBT_IP6=m
CONFIG_BRIDGE_EBT_LIMIT=m
CONFIG_BRIDGE_EBT_MARK=m
CONFIG_BRIDGE_EBT_PKTTYPE=m
CONFIG_BRIDGE_EBT_STP=m
CONFIG_BRIDGE_EBT_VLAN=m
CONFIG_BRIDGE_EBT_ARPREPLY=m
CONFIG_BRIDGE_EBT_DNAT=m
CONFIG_BRIDGE_EBT_MARK_T=m
CONFIG_BRIDGE_EBT_REDIRECT=m
CONFIG_BRIDGE_EBT_SNAT=m
CONFIG_BRIDGE_EBT_LOG=m
CONFIG_BRIDGE_EBT_ULOG=m
CONFIG_BRIDGE_EBT_NFLOG=m
CONFIG_IP_DCCP=m
CONFIG_INET_DCCP_DIAG=m
#
# DCCP CCIDs Configuration (EXPERIMENTAL)
#
# CONFIG_IP_DCCP_CCID2_DEBUG is not set
CONFIG_IP_DCCP_CCID3=y
# CONFIG_IP_DCCP_CCID3_DEBUG is not set
CONFIG_IP_DCCP_TFRC_LIB=y
#
# DCCP Kernel Hacking
#
# CONFIG_IP_DCCP_DEBUG is not set
# CONFIG_NET_DCCPPROBE is not set
CONFIG_IP_SCTP=m
# CONFIG_NET_SCTPPROBE is not set
# CONFIG_SCTP_DBG_MSG is not set
# CONFIG_SCTP_DBG_OBJCNT is not set
# CONFIG_SCTP_HMAC_NONE is not set
# CONFIG_SCTP_HMAC_SHA1 is not set
CONFIG_SCTP_HMAC_MD5=y
CONFIG_RDS=m
CONFIG_RDS_TCP=m
# CONFIG_RDS_DEBUG is not set
CONFIG_TIPC=m
CONFIG_TIPC_ADVANCED=y
CONFIG_TIPC_PORTS=8191
CONFIG_TIPC_LOG=0
# CONFIG_TIPC_DEBUG is not set
CONFIG_ATM=m
CONFIG_ATM_CLIP=m
# CONFIG_ATM_CLIP_NO_ICMP is not set
CONFIG_ATM_LANE=m
CONFIG_ATM_MPOA=m
CONFIG_ATM_BR2684=m
# CONFIG_ATM_BR2684_IPFILTER is not set
# CONFIG_L2TP is not set
CONFIG_STP=m
CONFIG_GARP=m
CONFIG_BRIDGE=m
CONFIG_BRIDGE_IGMP_SNOOPING=y
# CONFIG_NET_DSA is not set
CONFIG_VLAN_8021Q=m
CONFIG_VLAN_8021Q_GVRP=y
CONFIG_DECNET=m
# CONFIG_DECNET_ROUTER is not set
CONFIG_LLC=m
CONFIG_LLC2=m
CONFIG_IPX=m
# CONFIG_IPX_INTERN is not set
CONFIG_ATALK=m
CONFIG_DEV_APPLETALK=m
CONFIG_IPDDP=m
CONFIG_IPDDP_ENCAP=y
CONFIG_IPDDP_DECAP=y
CONFIG_X25=m
CONFIG_LAPB=m
CONFIG_ECONET=m
CONFIG_ECONET_AUNUDP=y
CONFIG_ECONET_NATIVE=y
CONFIG_WAN_ROUTER=m
CONFIG_PHONET=m
CONFIG_IEEE802154=m
# CONFIG_IEEE802154_6LOWPAN is not set
CONFIG_NET_SCHED=y
#
# Queueing/Scheduling
#
CONFIG_NET_SCH_CBQ=m
CONFIG_NET_SCH_HTB=m
CONFIG_NET_SCH_HFSC=m
CONFIG_NET_SCH_ATM=m
CONFIG_NET_SCH_PRIO=m
CONFIG_NET_SCH_MULTIQ=m
```

```
CONFIG_NET_SCH_RED=m
# CONFIG_NET_SCH_SFB is not set
CONFIG_NET_SCH_SFQ=m
CONFIG_NET_SCH_TEQL=m
CONFIG_NET_SCH_TBF=m
CONFIG_NET_SCH_GRED=m
CONFIG_NET_SCH_DSMARK=m
CONFIG_NET_SCH_NETEM=m
CONFIG_NET_SCH_DRR=m
# CONFIG_NET_SCH_MQPRIO is not set
# CONFIG_NET_SCH_CHOKE is not set
# CONFIG_NET_SCH_QFQ is not set
CONFIG_NET_SCH_INGRESS=m
#
# Classification
#
CONFIG_NET_CLS=y
CONFIG_NET_CLS_BASIC=m
CONFIG_NET_CLS_TCINDEX=m
CONFIG_NET_CLS_ROUTE4=m
CONFIG_NET_CLS_FW=m
CONFIG_NET_CLS_U32=m
CONFIG_CLS_U32_PERF=y
CONFIG_CLS_U32_MARK=y
CONFIG_NET_CLS_RSVP=m
CONFIG_NET_CLS_RSVP6=m
CONFIG_NET_CLS_FLOW=m
CONFIG_NET_CLS_CGROUP=y
CONFIG_NET_EMATCH=y
CONFIG_NET_EMATCH_STACK=32
CONFIG_NET_EMATCH_CMP=m
CONFIG_NET_EMATCH_NBYTE=m
CONFIG_NET_EMATCH_U32=m
CONFIG_NET_EMATCH_META=m
CONFIG_NET_EMATCH_TEXT=m
CONFIG_NET_CLS_ACT=y
CONFIG_NET_ACT_POLICE=m
CONFIG_NET_ACT_GACT=m
CONFIG_GACT_PROB=y
CONFIG_NET_ACT_MIRRED=m
CONFIG_NET_ACT_IPT=m
CONFIG_NET_ACT_NAT=m
CONFIG_NET_ACT_PEDIT=m
CONFIG_NET_ACT_SIMP=m
CONFIG_NET_ACT_SKBEDIT=m
# CONFIG_NET_ACT_CSUM is not set
CONFIG_NET_CLS_IND=y
CONFIG_NET_SCH_FIFO=y
CONFIG_DCB=y
CONFIG_DNS_RESOLVER=y
# CONFIG_BATMAN_ADV is not set
#
# Network testing
#
CONFIG_NET_PKTGEN=m
# CONFIG_NET_TCPPROBE is not set
CONFIG_NET_DROP_MONITOR=y
# CONFIG_HAMRADIO is not set
# CONFIG_CAN is not set
# CONFIG_IRDA is not set
# CONFIG_BT is not set
CONFIG_AF_RXRPC=m
# CONFIG_AF_RXRPC_DEBUG is not set
CONFIG_RXKAD=m
CONFIG_FIB_RULES=y
CONFIG_WIRELESS=y
# CONFIG_CFG80211 is not set
CONFIG_LIB80211=m
# CONFIG_LIB80211_DEBUG is not set
#
# CFG80211 needs to be enabled for MAC80211
#
# CONFIG_WIMAX is not set
# CONFIG_RFKILL is not set
# CONFIG_RFKILL_REGULATOR is not set
# CONFIG_NET_9P is not set
# CONFIG_CAIF is not set
# CONFIG_CEPH_LIB is not set
# CONFIG_NFC is not set
#
# Device Drivers
#
#
# Generic Driver Options
#
CONFIG_UEVENT_HELPER_PATH=""
CONFIG_DEVTMPFS=y
# CONFIG_DEVTMPFS_MOUNT is not set
CONFIG_STANDALONE=y
CONFIG_PREVENT_FIRMWARE_BUILD=y
CONFIG_FW_LOADER=y
```

```
# CONFIG_FIRMWARE_IN_KERNEL is not set
CONFIG_EXTRA_FIRMWARE=""
# CONFIG_DEBUG_DRIVER is not set
# CONFIG_DEBUG_DEVRES is not set
# CONFIG_SYS_HYPERVISOR is not set
CONFIG_REGMAP=y
CONFIG_REGMAP_I2C=m
CONFIG_CONNECTOR=y
CONFIG_PROC_EVENTS=y
CONFIG_MTD=y
# CONFIG_MTD_TESTS is not set
CONFIG_MTD_REDBOOT_PARTS=m
CONFIG_MTD_REDBOOT_DIRECTORY_BLOCK=-1
# CONFIG_MTD_REDBOOT_PARTS_UNALLOCATED is not set
# CONFIG_MTD_REDBOOT_PARTS_READONLY is not set
# CONFIG_MTD_CMDLINE_PARTS is not set
# CONFIG_MTD_OF_PARTS is not set
CONFIG_MTD_AR7_PARTS=m
#
# User Modules And Translation Layers
#
CONFIG_MTD_CHAR=m
CONFIG_MTD_BLKDEVS=m
CONFIG_MTD_BLOCK=m
CONFIG_MTD_BLOCK_RO=m
CONFIG_FTL=m
CONFIG_NFTL=m
CONFIG_NFTL_RW=y
CONFIG_INFTL=m
CONFIG_RFD_FTL=m
CONFIG_SSFDC=m
# CONFIG_SM_FTL is not set
CONFIG_MTD_OOPS=m
#
# RAM/ROM/Flash chip drivers
#
CONFIG_MTD_CFI=m
CONFIG_MTD_JEDECPROBE=m
CONFIG_MTD_GEN_PROBE=m
# CONFIG_MTD_CFI_ADV_OPTIONS is not set
CONFIG_MTD_MAP_BANK_WIDTH_1=y
CONFIG_MTD_MAP_BANK_WIDTH_2=y
CONFIG_MTD_MAP_BANK_WIDTH_4=y
# CONFIG_MTD_MAP_BANK_WIDTH_8 is not set
# CONFIG_MTD_MAP_BANK_WIDTH_16 is not set
# CONFIG_MTD_MAP_BANK_WIDTH_32 is not set
CONFIG_MTD_CFI_I1=y
CONFIG_MTD_CFI_I2=y
# CONFIG_MTD_CFI_I4 is not set
# CONFIG_MTD_CFI_I8 is not set
CONFIG_MTD_CFI_INTELEXT=m
CONFIG_MTD_CFI_AMDSTD=m
CONFIG_MTD_CFI_STAA=m
CONFIG_MTD_CFI_UTIL=m
CONFIG_MTD_RAM=m
CONFIG_MTD_ROM=m
CONFIG_MTD_ABSENT=m
#
# Mapping drivers for chip access
#
CONFIG_MTD_COMPLEX_MAPPINGS=y
CONFIG_MTD_PHYSMAP=m
# CONFIG_MTD_PHYSMAP_COMPAT is not set
# CONFIG_MTD_PHYSMAP_OF is not set
# CONFIG_MTD_GPIO_ADDR is not set
CONFIG_MTD_PLATRAM=m
# CONFIG_MTD_LATCH_ADDR is not set
#
# Self-contained MTD device drivers
#
CONFIG_MTD_DATAFLASH=m
# CONFIG_MTD_DATAFLASH_WRITE_VERIFY is not set
# CONFIG_MTD_DATAFLASH_OTP is not set
CONFIG_MTD_M25P80=m
CONFIG_M25PXX_USE_FAST_READ=y
CONFIG_MTD_SST25L=m
CONFIG_MTD_SLRAM=m
CONFIG_MTD_PHRAM=m
CONFIG_MTD_MTDRAM=m
CONFIG_MTDRAM_TOTAL_SIZE=4096
CONFIG_MTDRAM_ERASE_SIZE=128
CONFIG_MTD_BLOCK2MTD=m
#
# Disk-On-Chip Device Drivers
#
CONFIG_MTD_DOC2000=m
CONFIG_MTD_DOC2001=m
CONFIG_MTD_DOC2001PLUS=m
# CONFIG_MTD_DOCG3 is not set
CONFIG_MTD_DOCPROBE=m
CONFIG_MTD_DOCECC=m
```

```
# CONFIG_MTD_DOCPROBE_ADVANCED is not set
CONFIG_MTD_DOCPROBE_ADDRESS=0x0
CONFIG_MTD_NAND_ECC=m
# CONFIG_MTD_NAND_ECC_SMC is not set
CONFIG_MTD_NAND=m
# CONFIG_MTD_NAND_VERIFY_WRITE is not set
# CONFIG_MTD_NAND_ECC_BCH is not set
# CONFIG_MTD_SM_COMMON is not set
# CONFIG_MTD_NAND_MUSEUM_IDS is not set
CONFIG_MTD_NAND_IDS=m
# CONFIG_MTD_NAND_NDFC is not set
CONFIG_MTD_NAND_DISKONCHIP=m
# CONFIG_MTD_NAND_DISKONCHIP_PROBE_ADVANCED is not set
CONFIG_MTD_NAND_DISKONCHIP_PROBE_ADDRESS=0
# CONFIG_MTD_NAND_DISKONCHIP_BBTWRITE is not set
CONFIG_MTD_NAND_NANDSIM=m
CONFIG_MTD_NAND_PLATFORM=m
# CONFIG_MTD_NAND_FSL_ELBC is not set
CONFIG_MTD_ONENAND=m
CONFIG_MTD_ONENAND_VERIFY_WRITE=y
CONFIG_MTD_ONENAND_GENERIC=m
# CONFIG_MTD_ONENAND_OTP is not set
CONFIG_MTD_ONENAND_2X_PROGRAM=y
CONFIG_MTD_ONENAND_SIM=m
#
# LPDDR flash memory drivers
#
CONFIG_MTD_LPDDR=m
CONFIG_MTD_QINFO_PROBE=m
CONFIG_MTD_UBI=m
CONFIG_MTD_UBI_WL_THRESHOLD=4096
CONFIG_MTD_UBI_BEB_RESERVE=1
# CONFIG_MTD_UBI_GLUEBI is not set
# CONFIG_MTD_UBI_DEBUG is not set
CONFIG_DTC=y
CONFIG_OF=y
#
# Device Tree and Open Firmware support
#
# CONFIG_PROC_DEVICETREE is not set
CONFIG_OF_FLATTREE=y
CONFIG_OF_EARLY_FLATTREE=y
CONFIG_OF_DYNAMIC=y
CONFIG_OF_ADDRESS=y
CONFIG_OF_IRQ=y
CONFIG_OF_DEVICE=y
CONFIG_OF_GPIO=y
CONFIG_OF_I2C=m
CONFIG_OF_NET=y
CONFIG_OF_SPI=y
CONFIG_OF_MDIO=y
# CONFIG_PARPORT is not set
CONFIG_BLK_DEV=y
# CONFIG_BLK_DEV_FD is not set
# CONFIG_BLK_DEV_COW_COMMON is not set
CONFIG_BLK_DEV_LOOP=y
CONFIG_BLK_DEV_LOOP_MIN_COUNT=8
# CONFIG_BLK_DEV_CRYPTOLOOP is not set
CONFIG_BLK_DEV_DRBD=m
# CONFIG_DRBD_FAULT_INJECTION is not set
CONFIG_BLK_DEV_NBD=y
CONFIG_BLK_DEV_OSD=m
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_RAM_COUNT=16
CONFIG_BLK_DEV_RAM_SIZE=65536
# CONFIG_BLK_DEV_XIP is not set
# CONFIG_CDROM_PKTCDVD is not set
# CONFIG_ATA_OVER_ETH is not set
CONFIG_XILINX_SYSACE=y
# CONFIG_VIRTIO_BLK is not set
# CONFIG_BLK_DEV_HD is not set
# CONFIG_BLK_DEV_RBD is not set
# CONFIG_SENSORS_LIS3LV02D is not set
CONFIG_MISC_DEVICES=y
# CONFIG_AD525X_DPOT is not set
# CONFIG_ICS932S401 is not set
# CONFIG_ENCLOSURE_SERVICES is not set
# CONFIG_APDS9802ALS is not set
# CONFIG_ISL29003 is not set
# CONFIG_ISL29020 is not set
# CONFIG_SENSORS_TSL2550 is not set
# CONFIG_SENSORS_BH1780 is not set
# CONFIG_SENSORS_BH1770 is not set
# CONFIG_SENSORS_APDS990X is not set
# CONFIG_HMC6352 is not set
# CONFIG_DS1682 is not set
# CONFIG_TI_DAC7512 is not set
# CONFIG_BMP085 is not set
# CONFIG_USB_SWITCH_FSA9480 is not set
# CONFIG_C2PORT is not set
#
```

```
# EEPROM support
#
CONFIG_EEPROM_AT24=m
CONFIG_EEPROM_AT25=m
CONFIG_EEPROM_LEGACY=m
CONFIG_EEPROM_MAX6875=m
CONFIG_EEPROM_93CX6=m
# CONFIG_EEPROM_93XX46 is not set
#
# Texas Instruments shared transport line discipline
#
# CONFIG_TI_ST is not set
# CONFIG_SENSORS_LIS3_SPI is not set
# CONFIG_SENSORS_LIS3_I2C is not set
#
# Altera FPGA firmware download module
#
# CONFIG_ALTERA_STAPL is not set
CONFIG_HAVE_IDE=y
# CONFIG_IDE is not set
#
# SCSI device support
#
CONFIG_SCSI_MOD=m
CONFIG_RAID_ATTRS=m
CONFIG_SCSI=m
CONFIG_SCSI_DMA=y
CONFIG_SCSI_TGT=m
CONFIG_SCSI_NETLINK=y
CONFIG_SCSI_PROC_FS=y
#
# SCSI support type (disk, tape, CD-ROM)
#
CONFIG_BLK_DEV_SD=m
CONFIG_CHR_DEV_ST=m
CONFIG_CHR_DEV_OSST=m
CONFIG_BLK_DEV_SR=m
CONFIG_BLK_DEV_SR_VENDOR=y
CONFIG_CHR_DEV_SG=m
CONFIG_CHR_DEV_SCH=m
CONFIG_SCSI_MULTI_LUN=y
CONFIG_SCSI_CONSTANTS=y
CONFIG_SCSI_LOGGING=y
CONFIG_SCSI_SCAN_ASYNC=y
CONFIG_SCSI_WAIT_SCAN=m
#
# SCSI Transports
#
CONFIG_SCSI_SPI_ATTRS=m
CONFIG_SCSI_FC_ATTRS=m
CONFIG_SCSI_FC_TGT_ATTRS=y
CONFIG_SCSI_ISCSI_ATTRS=m
CONFIG_SCSI_SAS_ATTRS=m
CONFIG_SCSI_SAS_LIBSAS=m
CONFIG_SCSI_SAS_HOST_SMP=y
CONFIG_SCSI_SRP_ATTRS=m
CONFIG_SCSI_SRP_TGT_ATTRS=y
CONFIG_SCSI_LOWLEVEL=y
CONFIG_ISCSI_TCP=m
CONFIG_ISCSI_BOOT_SYSFS=m
CONFIG_LIBFC=m
CONFIG_LIBFCOE=m
CONFIG_SCSI_DEBUG=m
CONFIG_SCSI_DH=m
CONFIG_SCSI_DH_RDAC=m
CONFIG_SCSI_DH_HP_SW=m
CONFIG_SCSI_DH_EMC=m
CONFIG_SCSI_DH_ALUA=m
CONFIG_SCSI_OSD_INITIATOR=m
CONFIG_SCSI_OSD_ULD=m
CONFIG_SCSI_OSD_DPRINT_SENSE=1
# CONFIG_SCSI_OSD_DEBUG is not set
# CONFIG_ATA is not set
CONFIG_MD=y
CONFIG_BLK_DEV_MD=m
CONFIG_MD_LINEAR=m
CONFIG_MD_RAID0=m
CONFIG_MD_RAID1=m
CONFIG_MD_RAID10=m
CONFIG_MD_RAID456=m
CONFIG_MD_MULTIPATH=m
CONFIG_MD_FAULTY=m
CONFIG_BLK_DEV_DM=m
# CONFIG_DM_DEBUG is not set
CONFIG_DM_CRYPT=m
CONFIG_DM_SNAPSHOT=m
# CONFIG_DM_THIN_PROVISIONING is not set
CONFIG_DM_MIRROR=m
# CONFIG_DM_RAID is not set
CONFIG_DM_LOG_USERSPACE=m
CONFIG_DM_ZERO=m
```

```
CONFIG_DM_MULTIPATH=m
CONFIG_DM_MULTIPATH_QL=m
CONFIG_DM_MULTIPATH_ST=m
CONFIG_DM_DELAY=m
CONFIG_DM_UEVENT=y
# CONFIG_DM_FLAKEY is not set
# CONFIG_TARGET_CORE is not set
CONFIG_MACINTOSH_DRIVERS=y
CONFIG_MAC_EMUMOUSEBTN=y
# CONFIG_WINDFARM is not set
CONFIG_NETDEVICES=y
CONFIG_NET_CORE=y
CONFIG_BONDING=m
CONFIG_DUMMY=m
CONFIG_EQUALIZER=m
CONFIG_MII=m
CONFIG_IEEE802154_DRIVERS=m
CONFIG_IEEE802154_FAKEHARD=m
CONFIG_IFB=m
CONFIG_MACVLAN=m
CONFIG_MACVTAP=m
CONFIG_NETCONSOLE=m
CONFIG_NETCONSOLE_DYNAMIC=y
CONFIG_NETPOLL=y
# CONFIG_NETPOLL_TRAP is not set
CONFIG_NET_POLL_CONTROLLER=y
CONFIG_TUN=m
CONFIG_VETH=m
CONFIG_VIRTIO_NET=m
# CONFIG_ATM_DRIVERS is not set
#
# CAIF transport drivers
#
CONFIG_ETHERNET=y
# CONFIG_NET_VENDOR_BROADCOM is not set
# CONFIG_NET_VENDOR_CHELSIO is not set
# CONFIG_DNET is not set
# CONFIG_NET_VENDOR_IBM is not set
# CONFIG_NET_VENDOR_INTEL is not set
# CONFIG_NET_VENDOR_MARVELL is not set
# CONFIG_NET_VENDOR_MICREL is not set
# CONFIG_NET_VENDOR_MICROCHIP is not set
# CONFIG_NET_VENDOR_NATSEMI is not set
# CONFIG_ETHOC is not set
# CONFIG_NET_VENDOR_SEEQ is not set
# CONFIG_NET_VENDOR_STMICRO is not set
CONFIG_NET_VENDOR_XILINX=y
CONFIG_XILINX_EMACLITE=y
CONFIG_XILINX_LL_TEMAC=y
CONFIG_PHYLIB=y
#
# MII PHY device drivers
#
CONFIG_MARVELL_PHY=m
CONFIG_DAVICOM_PHY=m
CONFIG_QSEMI_PHY=m
CONFIG_LXT_PHY=m
CONFIG_CICADA_PHY=m
CONFIG_VITESSE_PHY=m
CONFIG_SMSC_PHY=m
CONFIG_BROADCOM_PHY=m
CONFIG_ICPLUS_PHY=m
CONFIG_REALTEK_PHY=m
CONFIG_NATIONAL_PHY=m
CONFIG_STE10XP=m
CONFIG_LSI_ET1011C_PHY=m
# CONFIG_MICREL_PHY is not set
# CONFIG_FIXED_PHY is not set
CONFIG_MDIO_BITBANG=m
# CONFIG_MDIO_GPIO is not set
CONFIG_PPP=m
CONFIG_PPP_BSDCOMP=m
CONFIG_PPP_DEFLATE=m
CONFIG_PPP_FILTER=y
CONFIG_PPP_MPPE=m
CONFIG_PPP_MULTILINK=y
CONFIG_PPPOATM=m
CONFIG_PPPOE=m
CONFIG_PPP_ASYNC=m
CONFIG_PPP_SYNC_TTY=m
CONFIG_SLIP=m
CONFIG_SLHC=m
CONFIG_SLIP_COMPRESSED=y
CONFIG_SLIP_SMART=y
CONFIG_SLIP_MODE_SLIP6=y
# CONFIG_WLAN is not set
#
# Enable WiMAX (Networking options) to see the WiMAX drivers
#
# CONFIG_WAN is not set
# CONFIG_ISDN is not set
```

```
# CONFIG_PHONE is not set
#
# Input device support
#
CONFIG_INPUT=y
CONFIG_INPUT_FF_MEMLESS=m
CONFIG_INPUT_POLLDEV=m
# CONFIG_INPUT_SPARSEKMAP is not set
#
# Userland interfaces
#
CONFIG_INPUT_MOUSEDEV=y
CONFIG_INPUT_MOUSEDEV_PSAUX=y
CONFIG_INPUT_MOUSEDEV_SCREEN_X=1024
CONFIG_INPUT_MOUSEDEV_SCREEN_Y=768
CONFIG_INPUT_JOYDEV=m
CONFIG_INPUT_EVDEV=m
# CONFIG_INPUT_EVBUG is not set
#
# Input Device Drivers
#
CONFIG_INPUT_KEYBOARD=y
CONFIG_KEYBOARD_ADP5588=m
# CONFIG_KEYBOARD_ADP5589 is not set
CONFIG_KEYBOARD_ATKBD=y
# CONFIG_KEYBOARD_QT1070 is not set
# CONFIG_KEYBOARD_QT2160 is not set
CONFIG_KEYBOARD_LKKBD=m
# CONFIG_KEYBOARD_GPIO is not set
# CONFIG_KEYBOARD_GPIO_POLLED is not set
# CONFIG_KEYBOARD_TCA6416 is not set
# CONFIG_KEYBOARD_MATRIX is not set
CONFIG_KEYBOARD_MAX7359=m
# CONFIG_KEYBOARD_MCS is not set
# CONFIG_KEYBOARD_MPR121 is not set
CONFIG_KEYBOARD_NEWTON=m
CONFIG_KEYBOARD_OPENCORES=m
CONFIG_KEYBOARD_STOWAWAY=m
CONFIG_KEYBOARD_SUNKBD=m
CONFIG_KEYBOARD_XTKBD=m
CONFIG_INPUT_MOUSE=y
CONFIG_MOUSE_PS2=m
CONFIG_MOUSE_PS2_ALPS=y
CONFIG_MOUSE_PS2_LOGIPS2PP=y
CONFIG_MOUSE_PS2_SYNAPTICS=y
CONFIG_MOUSE_PS2_TRACKPOINT=y
CONFIG_MOUSE_PS2_ELANTECH=y
CONFIG_MOUSE_PS2_SENTELIC=y
# CONFIG_MOUSE_PS2_TOUCHKIT is not set
CONFIG_MOUSE_SERIAL=m
CONFIG_MOUSE_VSXXXAA=m
# CONFIG_MOUSE_GPIO is not set
CONFIG_MOUSE_SYNAPTICS_I2C=m
CONFIG_INPUT_JOYSTICK=y
CONFIG_JOYSTICK_ANALOG=m
CONFIG_JOYSTICK_A3D=m
CONFIG_JOYSTICK_ADI=m
CONFIG_JOYSTICK_COBRA=m
CONFIG_JOYSTICK_GF2K=m
CONFIG_JOYSTICK_GRIP=m
CONFIG_JOYSTICK_GRIP_MP=m
CONFIG_JOYSTICK_GUILLEMOT=m
CONFIG_JOYSTICK_INTERACT=m
CONFIG_JOYSTICK_SIDEWINDER=m
CONFIG_JOYSTICK_TMDC=m
CONFIG_JOYSTICK_IFORCE=m
CONFIG_JOYSTICK_IFORCE_232=y
CONFIG_JOYSTICK_WARRIOR=m
CONFIG_JOYSTICK_MAGELLAN=m
CONFIG_JOYSTICK_SPACEORB=m
CONFIG_JOYSTICK_SPACEBALL=m
CONFIG_JOYSTICK_STINGER=m
CONFIG_JOYSTICK_TWIDJOY=m
CONFIG_JOYSTICK_ZHENHUA=m
# CONFIG_JOYSTICK_AS5011 is not set
CONFIG_JOYSTICK_JOYDUMP=m
CONFIG_INPUT_TABLET=y
CONFIG_INPUT_TOUCHSCREEN=y
CONFIG_TOUCHSCREEN_ADS7846=m
CONFIG_TOUCHSCREEN_AD7877=m
CONFIG_TOUCHSCREEN_AD7879=m
CONFIG_TOUCHSCREEN_AD7879_I2C=m
# CONFIG_TOUCHSCREEN_AD7879_SPI is not set
# CONFIG_TOUCHSCREEN_ATMEL_MXT is not set
# CONFIG_TOUCHSCREEN_BU21013 is not set
# CONFIG_TOUCHSCREEN_CY8CTMG110 is not set
# CONFIG_TOUCHSCREEN_DYNAPRO is not set
# CONFIG_TOUCHSCREEN_HAMPSHIRE is not set
CONFIG_TOUCHSCREEN_EETI=m
CONFIG_TOUCHSCREEN_FUJITSU=m
CONFIG_TOUCHSCREEN_GUNZE=m
```

```
CONFIG_TOUCHSCREEN_ELO=m
CONFIG_TOUCHSCREEN_WACOM_W8001=m
# CONFIG_TOUCHSCREEN_MAX11801 is not set
CONFIG_TOUCHSCREEN_MCS5000=m
CONFIG_TOUCHSCREEN_MTOUCH=m
CONFIG_TOUCHSCREEN_INEXIO=m
CONFIG_TOUCHSCREEN_MK712=m
CONFIG_TOUCHSCREEN_PENMOUNT=m
CONFIG_TOUCHSCREEN_TOUCHRIGHT=m
CONFIG_TOUCHSCREEN_TOUCHWIN=m
CONFIG_TOUCHSCREEN_TOUCHIT213=m
# CONFIG_TOUCHSCREEN_TSC_SERIO is not set
# CONFIG_TOUCHSCREEN_TSC2005 is not set
CONFIG_TOUCHSCREEN_TSC2007=m
# CONFIG_TOUCHSCREEN_ST1232 is not set
# CONFIG_TOUCHSCREEN_TPS6507X is not set
CONFIG_INPUT_MISC=y
# CONFIG_INPUT_AD714X is not set
# CONFIG_INPUT_BMA150 is not set
# CONFIG_INPUT_MMA8450 is not set
# CONFIG_INPUT_MPU3050 is not set
# CONFIG_INPUT_KXTJ9 is not set
CONFIG_INPUT_UINPUT=m
CONFIG_INPUT_PCF50633_PMU=m
# CONFIG_INPUT_PCF8574 is not set
# CONFIG_INPUT_GPIO_ROTARY_ENCODER is not set
# CONFIG_INPUT_ADXL34X is not set
# CONFIG_INPUT_CMA3000 is not set
#
# Hardware I/O ports
#
CONFIG_SERIO=y
# CONFIG_SERIO_I8042 is not set
CONFIG_SERIO_SERPORT=m
CONFIG_SERIO_LIBPS2=y
CONFIG_SERIO_RAW=m
# CONFIG_SERIO_XILINX_XPS_PS2 is not set
# CONFIG_SERIO_ALTERA_PS2 is not set
# CONFIG_SERIO_PS2MULT is not set
CONFIG_GAMEPORT=m
CONFIG_GAMEPORT_NS558=m
CONFIG_GAMEPORT_L4=m
#
# Character devices
#
CONFIG_VT=y
CONFIG_CONSOLE_TRANSLATIONS=y
CONFIG_VT_CONSOLE=y
CONFIG_VT_CONSOLE_SLEEP=y
CONFIG_HW_CONSOLE=y
CONFIG_VT_HW_CONSOLE_BINDING=y
CONFIG_UNIX98_PTYS=y
CONFIG_DEVPTS_MULTIPLE_INSTANCES=y
CONFIG_LEGACY_PTYS=y
CONFIG_LEGACY_PTY_COUNT=256
# CONFIG_SERIAL_NONSTANDARD is not set
# CONFIG_N_GSM is not set
# CONFIG_TRACE_SINK is not set
# CONFIG_PPC_EPAPR_HV_BYTECHAN is not set
# CONFIG_DEVKMEM is not set
#
# Serial drivers
#
CONFIG_SERIAL_8250=y
CONFIG_SERIAL_8250_CONSOLE=y
CONFIG_SERIAL_8250_NR_UARTS=32
CONFIG_SERIAL_8250_RUNTIME_UARTS=4
CONFIG_SERIAL_8250_EXTENDED=y
CONFIG_SERIAL_8250_MANY_PORTS=y
CONFIG_SERIAL_8250_SHARE_IRQ=y
# CONFIG_SERIAL_8250_DETECT_IRQ is not set
CONFIG_SERIAL_8250_RSA=y
# CONFIG_SERIAL_8250_DW is not set
#
# Non-8250 serial port support
#
CONFIG_SERIAL_MAX3100=m
# CONFIG_SERIAL_MAX3107 is not set
CONFIG_SERIAL_UARTLITE=y
CONFIG_SERIAL_UARTLITE_CONSOLE=y
CONFIG_SERIAL_CORE=y
CONFIG_SERIAL_CORE_CONSOLE=y
# CONFIG_SERIAL_OF_PLATFORM is not set
# CONFIG_SERIAL_OF_PLATFORM_NWPSERIAL is not set
# CONFIG_SERIAL_TIMBERDALE is not set
# CONFIG_SERIAL_ALTERA_JTAGUART is not set
# CONFIG_SERIAL_ALTERA_UART is not set
# CONFIG_SERIAL_IFX6X60 is not set
CONFIG_SERIAL_XILINX_PS_UART=y
CONFIG_SERIAL_XILINX_PS_UART_CONSOLE=y
# CONFIG_TTY_PRINTK is not set
```

```
CONFIG_HVC_DRIVER=y
# CONFIG_HVC_UDBG is not set
CONFIG_VIRTIO_CONSOLE=m
CONFIG_IPMI_HANDLER=m
# CONFIG_IPMI_PANIC_EVENT is not set
CONFIG_IPMI_DEVICE_INTERFACE=m
CONFIG_IPMI_SI=m
CONFIG_IPMI_WATCHDOG=m
CONFIG_IPMI_POWEROFF=m
CONFIG_HW_RANDOM=m
CONFIG_HW_RANDOM_TIMERIOMEM=m
CONFIG_HW_RANDOM_VIRTIO=m
# CONFIG_HW_RANDOM_PPC4XX is not set
CONFIG_NVRAM=m
CONFIG_XILINX_HWICAP=y
CONFIG_R3964=m
CONFIG_RAW_DRIVER=m
CONFIG_MAX_RAW_DEVS=256
CONFIG_TCG_TPM=m
CONFIG_TCG_TIS=m
CONFIG_TCG_NSC=m
CONFIG_TCG_ATMEL=m
# CONFIG_RAMOOPS is not set
CONFIG_I2C=m
CONFIG_I2C_BOARDINFO=y
CONFIG_I2C_COMPAT=y
CONFIG_I2C_CHARDEV=m
# CONFIG_I2C_MUX is not set
CONFIG_I2C_HELPER_AUTO=y
CONFIG_I2C_SMBUS=m
CONFIG_I2C_ALGOBIT=m
CONFIG_I2C_ALGOPCA=m
#
# I2C Hardware Bus support
#
#
# I2C system bus drivers (mostly embedded / system-on-chip)
#
# CONFIG_I2C_GPIO is not set
# CONFIG_I2C_IBM_IIC is not set
# CONFIG_I2C_MPC is not set
CONFIG_I2C_OCORES=m
CONFIG_I2C_PCA_PLATFORM=m
# CONFIG_I2C_PXA_PCI is not set
CONFIG_I2C_SIMTEC=m
# CONFIG_I2C_XILINX is not set
#
# External I2C/SMBus adapter drivers
#
CONFIG_I2C_PARPORT_LIGHT=m
CONFIG_I2C_TAOS_EVM=m
#
# Other I2C/SMBus bus drivers
#
CONFIG_I2C_STUB=m
# CONFIG_I2C_DEBUG_CORE is not set
# CONFIG_I2C_DEBUG_ALGO is not set
# CONFIG_I2C_DEBUG_BUS is not set
CONFIG_SPI=y
# CONFIG_SPI_DEBUG is not set
CONFIG_SPI_MASTER=y
#
# SPI Master Controller Drivers
#
# CONFIG_SPI_ALTERA is not set
CONFIG_SPI_BITBANG=m
# CONFIG_SPI_GPIO is not set
# CONFIG_SPI_OC_TINY is not set
# CONFIG_SPI_PPC4xx is not set
# CONFIG_SPI_PXA2XX_PCI is not set
# CONFIG_SPI_XILINX is not set
# CONFIG_SPI_DESIGNWARE is not set
#
# SPI Protocol Masters
#
# CONFIG_SPI_SPIDEV is not set
CONFIG_SPI_TLE62X0=m
#
# PPS support
#
CONFIG_PPS=m
# CONFIG_PPS_DEBUG is not set
#
# PPS clients support
#
# CONFIG_PPS_CLIENT_KTIMER is not set
# CONFIG_PPS_CLIENT_LDISC is not set
# CONFIG_PPS_CLIENT_GPIO is not set
#
# PPS generators support
#
```

```
#
# PTP clock support
#
# CONFIG_PTP_1588_CLOCK is not set
CONFIG_ARCH_WANT_OPTIONAL_GPIOLIB=y
CONFIG_GPIOLIB=y
# CONFIG_DEBUG_GPIO is not set
# CONFIG_GPIO_SYSFS is not set
#
# Memory mapped GPIO drivers:
#
# CONFIG_GPIO_GENERIC_PLATFORM is not set
# CONFIG_GPIO_IT8761E is not set
CONFIG_GPIO_XILINX=y
#
# I2C GPIO expanders:
#
# CONFIG_GPIO_MAX7300 is not set
# CONFIG_GPIO_MAX732X is not set
# CONFIG_GPIO_PCA953X is not set
# CONFIG_GPIO_PCF857X is not set
# CONFIG_GPIO_ADP5588 is not set
#
# PCI GPIO expanders:
#
#
# SPI GPIO expanders:
#
# CONFIG_GPIO_MAX7301 is not set
# CONFIG_GPIO_MCP23S08 is not set
# CONFIG_GPIO_MC33880 is not set
# CONFIG_GPIO_74X164 is not set
#
# AC97 GPIO expanders:
#
#
# MODULbus GPIO expanders:
#
CONFIG_W1=m
CONFIG_W1_CON=y
#
# 1-wire Bus Masters
#
CONFIG_W1_MASTER_DS2482=m
# CONFIG_W1_MASTER_DS1WM is not set
# CONFIG_W1_MASTER_GPIO is not set
#
# 1-wire Slaves
#
CONFIG_W1_SLAVE_THERM=m
CONFIG_W1_SLAVE_SMEM=m
# CONFIG_W1_SLAVE_DS2408 is not set
# CONFIG_W1_SLAVE_DS2423 is not set
CONFIG_W1_SLAVE_DS2431=m
CONFIG_W1_SLAVE_DS2433=m
# CONFIG_W1_SLAVE_DS2433_CRC is not set
CONFIG_W1_SLAVE_DS2760=m
# CONFIG_W1_SLAVE_DS2780 is not set
CONFIG_W1_SLAVE_BQ27000=m
CONFIG_POWER_SUPPLY=y
# CONFIG_POWER_SUPPLY_DEBUG is not set
CONFIG_PDA_POWER=m
# CONFIG_TEST_POWER is not set
CONFIG_BATTERY_DS2760=m
# CONFIG_BATTERY_DS2780 is not set
CONFIG_BATTERY_DS2782=m
# CONFIG_BATTERY_BQ20Z75 is not set
CONFIG_BATTERY_BQ27x00=m
CONFIG_BATTERY_BQ27X00_I2C=y
CONFIG_BATTERY_BQ27X00_PLATFORM=y
CONFIG_BATTERY_MAX17040=m
# CONFIG_BATTERY_MAX17042 is not set
CONFIG_CHARGER_PCF50633=m
# CONFIG_CHARGER_MAX8903 is not set
# CONFIG_CHARGER_GPIO is not set
CONFIG_HWMON=y
CONFIG_HWMON_VID=m
# CONFIG_HWMON_DEBUG_CHIP is not set
#
# Native drivers
#
# CONFIG_SENSORS_AD7314 is not set
CONFIG_SENSORS_AD7414=m
CONFIG_SENSORS_AD7418=m
CONFIG_SENSORS_ADCXX=m
CONFIG_SENSORS_ADM1021=m
CONFIG_SENSORS_ADM1025=m
CONFIG_SENSORS_ADM1026=m
CONFIG_SENSORS_ADM1029=m
CONFIG_SENSORS_ADM1031=m
CONFIG_SENSORS_ADM9240=m
```

```
# CONFIG_SENSORS_ADT7411 is not set
CONFIG_SENSORS_ADT7462=m
CONFIG_SENSORS_ADT7470=m
CONFIG_SENSORS_ADT7475=m
# CONFIG_SENSORS_ASC7621 is not set
CONFIG_SENSORS_ATXP1=m
# CONFIG_SENSORS_DS620 is not set
CONFIG_SENSORS_DS1621=m
CONFIG_SENSORS_F75375S=m
CONFIG_SENSORS_G760A=m
CONFIG_SENSORS_GL518SM=m
CONFIG_SENSORS_GL520SM=m
# CONFIG_SENSORS_GPIO_FAN is not set
CONFIG_SENSORS_IBMAEM=m
CONFIG_SENSORS_IBMPEX=m
# CONFIG_SENSORS_JC42 is not set
# CONFIG_SENSORS_LINEAGE is not set
CONFIG_SENSORS_LM63=m
CONFIG_SENSORS_LM70=m
# CONFIG_SENSORS_LM73 is not set
CONFIG_SENSORS_LM75=m
CONFIG_SENSORS_LM77=m
CONFIG_SENSORS_LM78=m
CONFIG_SENSORS_LM80=m
CONFIG_SENSORS_LM83=m
CONFIG_SENSORS_LM85=m
CONFIG_SENSORS_LM87=m
CONFIG_SENSORS_LM90=m
CONFIG_SENSORS_LM92=m
CONFIG_SENSORS_LM93=m
# CONFIG_SENSORS_LTC4151 is not set
CONFIG_SENSORS_LTC4215=m
CONFIG_SENSORS_LTC4245=m
# CONFIG_SENSORS_LTC4261 is not set
CONFIG_SENSORS_LM95241=m
# CONFIG_SENSORS_LM95245 is not set
CONFIG_SENSORS_MAX1111=m
# CONFIG_SENSORS_MAX16065 is not set
CONFIG_SENSORS_MAX1619=m
# CONFIG_SENSORS_MAX1668 is not set
# CONFIG_SENSORS_MAX6639 is not set
# CONFIG_SENSORS_MAX6642 is not set
CONFIG_SENSORS_MAX6650=m
# CONFIG_SENSORS_NTC_THERMISTOR is not set
CONFIG_SENSORS_PCF8591=m
# CONFIG_PMBUS is not set
# CONFIG_SENSORS_SHT15 is not set
# CONFIG_SENSORS_SHT21 is not set
# CONFIG_SENSORS_SMM665 is not set
# CONFIG_SENSORS_EMC1403 is not set
# CONFIG_SENSORS_EMC2103 is not set
# CONFIG_SENSORS_EMC6W201 is not set
CONFIG_SENSORS_SMSC47M192=m
# CONFIG_SENSORS_SCH56XX_COMMON is not set
# CONFIG_SENSORS_ADS1015 is not set
CONFIG_SENSORS_ADS7828=m
# CONFIG_SENSORS_ADS7871 is not set
# CONFIG_SENSORS_AMC6821 is not set
CONFIG_SENSORS_THMC50=m
# CONFIG_SENSORS_TMP102 is not set
CONFIG_SENSORS_TMP401=m
CONFIG_SENSORS_TMP421=m
CONFIG_SENSORS_W83781D=m
CONFIG_SENSORS_W83791D=m
CONFIG_SENSORS_W83792D=m
CONFIG_SENSORS_W83793=m
# CONFIG_SENSORS_W83795 is not set
CONFIG_SENSORS_W83L785TS=m
CONFIG_SENSORS_W83L786NG=m
CONFIG_THERMAL=m
CONFIG_THERMAL_HWMON=y
CONFIG_WATCHDOG=y
# CONFIG_WATCHDOG_CORE is not set
# CONFIG_WATCHDOG_NOWAYOUT is not set
#
# Watchdog Device Drivers
#
CONFIG_SOFT_WATCHDOG=m
# CONFIG_BOOKE_WDT is not set
CONFIG_SSB_POSSIBLE=y
#
# Sonics Silicon Backplane
#
CONFIG_SSB=m
# CONFIG_SSB_SILENT is not set
# CONFIG_SSB_DEBUG is not set
CONFIG_BCMA_POSSIBLE=y
#
# Broadcom specific AMBA
#
# CONFIG_BCMA is not set
```

```
#
# Multifunction device drivers
#
CONFIG_MFD_CORE=m
CONFIG_MFD_SM501=m
# CONFIG_MFD_SM501_GPIO is not set
CONFIG_HTC_PASIC3=m
# CONFIG_TPS6105X is not set
# CONFIG_TPS65010 is not set
# CONFIG_TPS6507X is not set
# CONFIG_MFD_TPS65912_SPI is not set
# CONFIG_MFD_TMIO is not set
CONFIG_MFD_WM8400=m
# CONFIG_MFD_WM831X_SPI is not set
CONFIG_MFD_PCF50633=m
CONFIG_PCF50633_ADC=m
CONFIG_PCF50633_GPIO=m
# CONFIG_MFD_MC13XXX is not set
# CONFIG_ABX500_CORE is not set
# CONFIG_EZX_PCAP is not set
# CONFIG_MFD_WL1273_CORE is not set
CONFIG_REGULATOR=y
# CONFIG_REGULATOR_DEBUG is not set
# CONFIG_REGULATOR_DUMMY is not set
CONFIG_REGULATOR_FIXED_VOLTAGE=m
# CONFIG_REGULATOR_VIRTUAL_CONSUMER is not set
CONFIG_REGULATOR_USERSPACE_CONSUMER=m
# CONFIG_REGULATOR_GPIO is not set
CONFIG_REGULATOR_BQ24022=m
CONFIG_REGULATOR_MAX1586=m
# CONFIG_REGULATOR_MAX8649 is not set
# CONFIG_REGULATOR_MAX8660 is not set
# CONFIG_REGULATOR_MAX8952 is not set
CONFIG_REGULATOR_WM8400=m
CONFIG_REGULATOR_PCF50633=m
CONFIG_REGULATOR_LP3971=m
# CONFIG_REGULATOR_LP3972 is not set
CONFIG_REGULATOR_TPS65023=m
CONFIG_REGULATOR_TPS6507X=m
# CONFIG_REGULATOR_ISL6271A is not set
# CONFIG_REGULATOR_AD5398 is not set
# CONFIG_REGULATOR_TPS6524X is not set
# CONFIG_MEDIA_SUPPORT is not set
#
# Graphics support
#
CONFIG_DRM=m
CONFIG_VGASTATE=m
CONFIG_VIDEO_OUTPUT_CONTROL=m
CONFIG_FB=y
CONFIG_FIRMWARE_EDID=y
# CONFIG_FB_DDC is not set
# CONFIG_FB_BOOT_VESA_SUPPORT is not set
CONFIG_FB_CFB_FILLRECT=m
CONFIG_FB_CFB_COPYAREA=m
CONFIG_FB_CFB_IMAGEBLIT=m
# CONFIG_FB_CFB_REV_PIXELS_IN_BYTE is not set
CONFIG_FB_SYS_FILLRECT=m
CONFIG_FB_SYS_COPYAREA=m
CONFIG_FB_SYS_IMAGEBLIT=m
CONFIG_FB_FOREIGN_ENDIAN=y
CONFIG_FB_BOTH_ENDIAN=y
# CONFIG_FB_BIG_ENDIAN is not set
# CONFIG_FB_LITTLE_ENDIAN is not set
CONFIG_FB_SYS_FOPS=m
# CONFIG_FB_WMT_GE_ROPS is not set
CONFIG_FB_DEFERRED_IO=y
# CONFIG_FB_SVGALIB is not set
# CONFIG_FB_MACMODES is not set
# CONFIG_FB_BACKLIGHT is not set
CONFIG_FB_MODE_HELPERS=y
CONFIG_FB_TILEBLITTING=y
#
# Frame buffer hardware drivers
#
# CONFIG_FB_OF is not set
CONFIG_FB_VGA16=m
CONFIG_FB_UVESA=m
CONFIG_FB_S1D13XXX=m
# CONFIG_FB_TMIO is not set
CONFIG_FB_SM501=m
# CONFIG_FB_IBM_GXT4500 is not set
# CONFIG_FB_XILINX is not set
CONFIG_FB_VIRTUAL=m
CONFIG_FB_METRONOME=m
CONFIG_FB_MB862XX=m
CONFIG_FB_MB862XX_LIME=y
CONFIG_FB_MB862XX_I2C=y
# CONFIG_FB_PRE_INIT_FB is not set
# CONFIG_FB_BROADSHEET is not set
CONFIG_BACKLIGHT_LCD_SUPPORT=y
```

```
# CONFIG_LCD_CLASS_DEVICE is not set
CONFIG_BACKLIGHT_CLASS_DEVICE=y
# CONFIG_BACKLIGHT_GENERIC is not set
# CONFIG_BACKLIGHT_ADP8860 is not set
# CONFIG_BACKLIGHT_ADP8870 is not set
# CONFIG_BACKLIGHT_PCF50633 is not set
#
# Display device support
#
CONFIG_DISPLAY_SUPPORT=m
#
# Display hardware drivers
#
#
# Console display driver support
#
CONFIG_DUMMY_CONSOLE=y
CONFIG_FRAMEBUFFER_CONSOLE=y
CONFIG_FRAMEBUFFER_CONSOLE_DETECT_PRIMARY=y
CONFIG_FRAMEBUFFER_CONSOLE_ROTATION=y
# CONFIG_FONTS is not set
CONFIG_FONT_8x8=y
CONFIG_FONT_8x16=y
# CONFIG_LOGO is not set
# CONFIG_SOUND is not set
CONFIG_HID_SUPPORT=y
CONFIG_HID=m
CONFIG_HIDRAW=y
CONFIG_HID_PID=y
#
# Special HID drivers
#
CONFIG_USB_SUPPORT=y
# CONFIG_USB_ARCH_HAS_HCD is not set
# CONFIG_USB_ARCH_HAS_OHCI is not set
# CONFIG_USB_ARCH_HAS_EHCI is not set
# CONFIG_USB_ARCH_HAS_XHCI is not set
# CONFIG_USB_OTG_WHITELIST is not set
# CONFIG_USB_OTG_BLACKLIST_HUB is not set
#
# NOTE: USB_STORAGE depends on SCSI but BLK_DEV_SD may
#
# CONFIG_USB_GADGET is not set
#
# OTG and related infrastructure
#
# CONFIG_MMC is not set
# CONFIG_MEMSTICK is not set
# CONFIG_NEW_LEDS is not set
# CONFIG_ACCESSIBILITY is not set
CONFIG_EDAC=y
#
# Reporting subsystems
#
# CONFIG_EDAC_DEBUG is not set
CONFIG_EDAC_MM_EDAC=m
# CONFIG_EDAC_PPC4XX is not set
CONFIG_RTC_LIB=y
CONFIG_RTC_CLASS=y
CONFIG_RTC_HCTOSYS=y
CONFIG_RTC_HCTOSYS_DEVICE="rtc0"
# CONFIG_RTC_DEBUG is not set
#
# RTC interfaces
#
CONFIG_RTC_INTF_SYSFS=y
CONFIG_RTC_INTF_PROC=y
CONFIG_RTC_INTF_DEV=y
# CONFIG_RTC_INTF_DEV_UIE_EMUL is not set
# CONFIG_RTC_DRV_TEST is not set
#
# I2C RTC drivers
#
CONFIG_RTC_DRV_DS1307=m
CONFIG_RTC_DRV_DS1374=m
CONFIG_RTC_DRV_DS1672=m
# CONFIG_RTC_DRV_DS3232 is not set
CONFIG_RTC_DRV_MAX6900=m
CONFIG_RTC_DRV_RS5C372=m
CONFIG_RTC_DRV_ISL1208=m
# CONFIG_RTC_DRV_ISL12022 is not set
CONFIG_RTC_DRV_X1205=m
CONFIG_RTC_DRV_PCF8563=m
CONFIG_RTC_DRV_PCF8583=m
CONFIG_RTC_DRV_M41T80=m
# CONFIG_RTC_DRV_M41T80_WDT is not set
# CONFIG_RTC_DRV_BQ32K is not set
CONFIG_RTC_DRV_S35390A=m
CONFIG_RTC_DRV_FM3130=m
CONFIG_RTC_DRV_RX8581=m
CONFIG_RTC_DRV_RX8025=m
```

```
# CONFIG_RTC_DRV_EM3027 is not set
# CONFIG_RTC_DRV_RV3029C2 is not set
#
# SPI RTC drivers
#
# CONFIG_RTC_DRV_M41T93 is not set
CONFIG_RTC_DRV_M41T94=m
CONFIG_RTC_DRV_DS1305=m
CONFIG_RTC_DRV_DS1390=m
CONFIG_RTC_DRV_MAX6902=m
CONFIG_RTC_DRV_R9701=m
CONFIG_RTC_DRV_RS5C348=m
CONFIG_RTC_DRV_DS3234=m
CONFIG_RTC_DRV_PCF2123=m
#
# Platform RTC drivers
#
CONFIG_RTC_DRV_CMOS=y
CONFIG_RTC_DRV_DS1286=m
CONFIG_RTC_DRV_DS1511=m
CONFIG_RTC_DRV_DS1553=m
CONFIG_RTC_DRV_DS1742=m
CONFIG_RTC_DRV_STK17TA8=m
CONFIG_RTC_DRV_M48T86=m
CONFIG_RTC_DRV_M48T35=m
CONFIG_RTC_DRV_M48T59=m
# CONFIG_RTC_DRV_MSM6242 is not set
CONFIG_RTC_DRV_BQ4802=m
# CONFIG_RTC_DRV_RP5C01 is not set
CONFIG_RTC_DRV_V3020=m
CONFIG_RTC_DRV_PCF50633=m
#
# on-CPU RTC drivers
#
# CONFIG_RTC_DRV_GENERIC is not set
CONFIG_DMADEVICES=y
# CONFIG_DMADEVICES_DEBUG is not set
#
# DMA Devices
#
# CONFIG_TIMB_DMA is not set
# CONFIG_AUXDISPLAY is not set
CONFIG_UIO=m
CONFIG_UIO_PDRV=m
CONFIG_UIO_PDRV_GENIRQ=m
CONFIG_VIRTIO=m
CONFIG_VIRTIO_RING=m
#
# Virtio drivers
#
CONFIG_VIRTIO_BALLOON=m
# CONFIG_VIRTIO_MMIO is not set
CONFIG_STAGING=y
# CONFIG_ECHO is not set
CONFIG_POHMELFS=m
# CONFIG_POHMELFS_DEBUG is not set
# CONFIG_IIO is not set
# CONFIG_XVMALLOC is not set
# CONFIG_ZRAM is not set
# CONFIG_FB_SM7XX is not set
# CONFIG_FT1000 is not set
#
# Speakup console speech
#
CONFIG_SPEAKUP=m
CONFIG_SPEAKUP_SYNTH_ACNTSA=m
CONFIG_SPEAKUP_SYNTH_ACNTPC=m
CONFIG_SPEAKUP_SYNTH_APOLLO=m
CONFIG_SPEAKUP_SYNTH_AUDPTR=m
CONFIG_SPEAKUP_SYNTH_BNS=m
CONFIG_SPEAKUP_SYNTH_DECTLK=m
CONFIG_SPEAKUP_SYNTH_DECEXT=m
# CONFIG_SPEAKUP_SYNTH_DECPC is not set
CONFIG_SPEAKUP_SYNTH_DTLK=m
CONFIG_SPEAKUP_SYNTH_KEYPC=m
CONFIG_SPEAKUP_SYNTH_LTLK=m
CONFIG_SPEAKUP_SYNTH_SOFT=m
CONFIG_SPEAKUP_SYNTH_SPKOUT=m
CONFIG_SPEAKUP_SYNTH_TXPRT=m
CONFIG_SPEAKUP_SYNTH_DUMMY=m
# CONFIG_TOUCHSCREEN_CLEARPAD_TM1217 is not set
# CONFIG_TOUCHSCREEN_SYNAPTICS_I2C_RMI4 is not set
# CONFIG_STAGING_MEDIA is not set
#
# Hardware Spinlock drivers
#
CONFIG_IOMMU_SUPPORT=y
# CONFIG_VIRT_DRIVERS is not set
# CONFIG_PM_DEVFREQ is not set
#
# File systems
```

```
#
CONFIG_EXT2_FS=y
CONFIG_EXT2_FS_XATTR=y
CONFIG_EXT2_FS_POSIX_ACL=y
CONFIG_EXT2_FS_SECURITY=y
CONFIG_EXT2_FS_XIP=y
# CONFIG_EXT3_FS is not set
# CONFIG_EXT4_FS is not set
CONFIG_FS_XIP=y
CONFIG_FS_MBCACHE=y
# CONFIG_REISERFS_FS is not set
# CONFIG_JFS_FS is not set
# CONFIG_XFS_FS is not set
# CONFIG_GFS2_FS is not set
# CONFIG_OCFS2_FS is not set
# CONFIG_BTRFS_FS is not set
# CONFIG_NILFS2_FS is not set
CONFIG_FS_POSIX_ACL=y
CONFIG_EXPORTFS=m
CONFIG_FILE_LOCKING=y
CONFIG_FSNOTIFY=y
CONFIG_DNOTIFY=y
CONFIG_INOTIFY_USER=y
# CONFIG_FANOTIFY is not set
CONFIG_QUOTA=y
CONFIG_QUOTA_NETLINK_INTERFACE=y
CONFIG_PRINT_QUOTA_WARNING=y
# CONFIG_QUOTA_DEBUG is not set
CONFIG_QUOTA_TREE=m
CONFIG_QFMT_V1=m
CONFIG_QFMT_V2=m
CONFIG_QUOTACTL=y
CONFIG_AUTOFS4_FS=m
CONFIG_FUSE_FS=m
CONFIG_CUSE=m
CONFIG_GENERIC_ACL=y
#
# Caches
#
CONFIG_FSCACHE=m
CONFIG_FSCACHE_STATS=y
# CONFIG_FSCACHE_HISTOGRAM is not set
# CONFIG_FSCACHE_DEBUG is not set
# CONFIG_FSCACHE_OBJECT_LIST is not set
CONFIG_CACHEFILES=m
# CONFIG_CACHEFILES_DEBUG is not set
# CONFIG_CACHEFILES_HISTOGRAM is not set
#
# CD-ROM/DVD Filesystems
#
CONFIG_ISO9660_FS=m
CONFIG_JOLIET=y
CONFIG_ZISOFS=y
CONFIG_UDF_FS=m
CONFIG_UDF_NLS=y
#
# DOS/FAT/NT Filesystems
#
CONFIG_FAT_FS=m
CONFIG_MSDOS_FS=m
CONFIG_VFAT_FS=m
CONFIG_FAT_DEFAULT_CODEPAGE=437
CONFIG_FAT_DEFAULT_IOCHARSET="utf8"
CONFIG_NTFS_FS=m
# CONFIG_NTFS_DEBUG is not set
CONFIG_NTFS_RW=y
#
# Pseudo filesystems
#
CONFIG_PROC_FS=y
CONFIG_PROC_KCORE=y
CONFIG_PROC_SYSCTL=y
CONFIG_PROC_PAGE_MONITOR=y
CONFIG_SYSFS=y
CONFIG_TMPFS=y
CONFIG_TMPFS_POSIX_ACL=y
CONFIG_TMPFS_XATTR=y
# CONFIG_HUGETLB_PAGE is not set
CONFIG_CONFIGFS_FS=m
CONFIG_MISC_FILESYSTEMS=y
CONFIG_ADFS_FS=m
# CONFIG_ADFS_FS_RW is not set
CONFIG_AFFS_FS=m
CONFIG_ECRYPT_FS=m
CONFIG_HFS_FS=m
CONFIG_HFSPLUS_FS=m
CONFIG_BEFS_FS=m
# CONFIG_BEFS_DEBUG is not set
CONFIG_BFS_FS=m
CONFIG_EFS_FS=m
CONFIG_JFFS2_FS=m
```

```
CONFIG_JFFS2_FS_DEBUG=0
CONFIG_JFFS2_FS_WRITEBUFFER=y
# CONFIG_JFFS2_FS_WBUF_VERIFY is not set
CONFIG_JFFS2_SUMMARY=y
CONFIG_JFFS2_FS_XATTR=y
CONFIG_JFFS2_FS_POSIX_ACL=y
CONFIG_JFFS2_FS_SECURITY=y
CONFIG_JFFS2_COMPRESSION_OPTIONS=y
CONFIG_JFFS2_ZLIB=y
CONFIG_JFFS2_LZO=y
CONFIG_JFFS2_RTIME=y
# CONFIG_JFFS2_RUBIN is not set
# CONFIG_JFFS2_CMODE_NONE is not set
CONFIG_JFFS2_CMODE_PRIORITY=y
# CONFIG_JFFS2_CMODE_SIZE is not set
# CONFIG_JFFS2_CMODE_FAVOURLZO is not set
CONFIG_UBIFS_FS=m
CONFIG_UBIFS_FS_XATTR=y
CONFIG_UBIFS_FS_ADVANCED_COMPR=y
CONFIG_UBIFS_FS_LZO=y
CONFIG_UBIFS_FS_ZLIB=y
# CONFIG_UBIFS_FS_DEBUG is not set
# CONFIG_LOGFS is not set
CONFIG_CRAMFS=m
CONFIG_SQUASHFS=m
# CONFIG_SQUASHFS_XATTR is not set
CONFIG_SQUASHFS_ZLIB=y
# CONFIG_SQUASHFS_LZO is not set
# CONFIG_SQUASHFS_XZ is not set
# CONFIG_SQUASHFS_4K_DEVBLK_SIZE is not set
# CONFIG_SQUASHFS_EMBEDDED is not set
CONFIG_SQUASHFS_FRAGMENT_CACHE_SIZE=3
CONFIG_VXFS_FS=m
CONFIG_MINIX_FS=m
CONFIG_OMFS_FS=m
CONFIG_HPFS_FS=m
CONFIG_QNX4FS_FS=m
CONFIG_ROMFS_FS=m
# CONFIG_ROMFS_BACKED_BY_BLOCK is not set
# CONFIG_ROMFS_BACKED_BY_MTD is not set
CONFIG_ROMFS_BACKED_BY_BOTH=y
CONFIG_ROMFS_ON_BLOCK=y
CONFIG_ROMFS_ON_MTD=y
# CONFIG_PSTORE is not set
CONFIG_SYSV_FS=m
CONFIG_UFS_FS=m
# CONFIG_UFS_FS_WRITE is not set
# CONFIG_UFS_DEBUG is not set
CONFIG_ORE=m
CONFIG_EXOFS_FS=m
# CONFIG_EXOFS_DEBUG is not set
CONFIG_NETWORK_FILESYSTEMS=y
CONFIG_NFS_FS=y
CONFIG_NFS_V3=y
CONFIG_NFS_V3_ACL=y
CONFIG_NFS_V4=y
# CONFIG_NFS_V4_1 is not set
CONFIG_ROOT_NFS=y
# CONFIG_NFS_USE_LEGACY_DNS is not set
CONFIG_NFS_USE_KERNEL_DNS=y
# CONFIG_NFS_USE_NEW_IDMAPPER is not set
CONFIG_NFSD=m
CONFIG_NFSD_V2_ACL=y
CONFIG_NFSD_V3=y
CONFIG_NFSD_V3_ACL=y
CONFIG_NFSD_V4=y
CONFIG_LOCKD=y
CONFIG_LOCKD_V4=y
CONFIG_NFS_ACL_SUPPORT=y
CONFIG_NFS_COMMON=y
CONFIG_SUNRPC=y
CONFIG_SUNRPC_GSS=y
CONFIG_RPCSEC_GSS_KRB5=m
# CONFIG_CEPH_FS is not set
CONFIG_CIFS=m
# CONFIG_CIFS_STATS is not set
CONFIG_CIFS_WEAK_PW_HASH=y
CONFIG_CIFS_UPCALL=y
CONFIG_CIFS_XATTR=y
CONFIG_CIFS_POSIX=y
# CONFIG_CIFS_DEBUG2 is not set
CONFIG_CIFS_DFS_UPCALL=y
# CONFIG_CIFS_FSCACHE is not set
# CONFIG_CIFS_ACL is not set
CONFIG_NCP_FS=m
CONFIG_NCPFS_PACKET_SIGNING=y
CONFIG_NCPFS_IOCTL_LOCKING=y
CONFIG_NCPFS_STRONG=y
CONFIG_NCPFS_NFS_NS=y
CONFIG_NCPFS_OS2_NS=y
# CONFIG_NCPFS_SMALLDOS is not set
```

```
CONFIG_NCPFS_NLS=y
CONFIG_NCPFS_EXTRAS=y
CONFIG_CODA_FS=m
CONFIG_AFS_FS=m
# CONFIG_AFS_DEBUG is not set
CONFIG_AFS_FSCACHE=y
#
# Partition Types
#
CONFIG_PARTITION_ADVANCED=y
CONFIG_ACORN_PARTITION=y
# CONFIG_ACORN_PARTITION_CUMANA is not set
# CONFIG_ACORN_PARTITION_EESOX is not set
CONFIG_ACORN_PARTITION_ICS=y
# CONFIG_ACORN_PARTITION_ADFS is not set
# CONFIG_ACORN_PARTITION_POWERTEC is not set
CONFIG_ACORN_PARTITION_RISCIX=y
CONFIG_OSF_PARTITION=y
CONFIG_AMIGA_PARTITION=y
CONFIG_ATARI_PARTITION=y
CONFIG_MAC_PARTITION=y
CONFIG_MSDOS_PARTITION=y
CONFIG_BSD_DISKLABEL=y
CONFIG_MINIX_SUBPARTITION=y
CONFIG_SOLARIS_X86_PARTITION=y
CONFIG_UNIXWARE_DISKLABEL=y
CONFIG_LDM_PARTITION=y
# CONFIG_LDM_DEBUG is not set
CONFIG_SGI_PARTITION=y
CONFIG_ULTRIX_PARTITION=y
CONFIG_SUN_PARTITION=y
CONFIG_KARMA_PARTITION=y
CONFIG_EFI_PARTITION=y
# CONFIG_SYSV68_PARTITION is not set
CONFIG_NLS=m
CONFIG_NLS_DEFAULT="utf8"
CONFIG_NLS_CODEPAGE_437=m
CONFIG_NLS_CODEPAGE_737=m
CONFIG_NLS_CODEPAGE_775=m
CONFIG_NLS_CODEPAGE_850=m
CONFIG_NLS_CODEPAGE_852=m
CONFIG_NLS_CODEPAGE_855=m
CONFIG_NLS_CODEPAGE_857=m
CONFIG_NLS_CODEPAGE_860=m
CONFIG_NLS_CODEPAGE_861=m
CONFIG_NLS_CODEPAGE_862=m
CONFIG_NLS_CODEPAGE_863=m
CONFIG_NLS_CODEPAGE_864=m
CONFIG_NLS_CODEPAGE_865=m
CONFIG_NLS_CODEPAGE_866=m
CONFIG_NLS_CODEPAGE_869=m
CONFIG_NLS_CODEPAGE_936=m
CONFIG_NLS_CODEPAGE_950=m
CONFIG_NLS_CODEPAGE_932=m
CONFIG_NLS_CODEPAGE_949=m
CONFIG_NLS_CODEPAGE_874=m
CONFIG_NLS_ISO8859_8=m
CONFIG_NLS_CODEPAGE_1250=m
CONFIG_NLS_CODEPAGE_1251=m
CONFIG_NLS_ASCII=m
CONFIG_NLS_ISO8859_1=m
CONFIG_NLS_ISO8859_2=m
CONFIG_NLS_ISO8859_3=m
CONFIG_NLS_ISO8859_4=m
CONFIG_NLS_ISO8859_5=m
CONFIG_NLS_ISO8859_6=m
CONFIG_NLS_ISO8859_7=m
CONFIG_NLS_ISO8859_9=m
CONFIG_NLS_ISO8859_13=m
CONFIG_NLS_ISO8859_14=m
CONFIG_NLS_ISO8859_15=m
CONFIG_NLS_KOI8_R=m
CONFIG_NLS_KOI8_U=m
CONFIG_NLS_UTF8=m
CONFIG_DLM=m
CONFIG_DLM_DEBUG=y
CONFIG_BINARY_PRINTF=y
#
# Library routines
#
CONFIG_RAID6_PQ=m
CONFIG_BITREVERSE=y
CONFIG_CRC_CCITT=m
CONFIG_CRC16=m
CONFIG_CRC_T10DIF=m
CONFIG_CRC_ITU_T=m
CONFIG_CRC32=y
CONFIG_CRC7=m
CONFIG_LIBCRC32C=m
# CONFIG_CRC8 is not set
CONFIG_ZLIB_INFLATE=y
```

```
CONFIG_ZLIB_DEFLATE=m
CONFIG_LZO_COMPRESS=y
CONFIG_LZO_DECOMPRESS=y
CONFIG_XZ_DEC=y
CONFIG_XZ_DEC_X86=y
CONFIG_XZ_DEC_POWERPC=y
CONFIG_XZ_DEC_IA64=y
CONFIG_XZ_DEC_ARM=y
CONFIG_XZ_DEC_ARMTHUMB=y
CONFIG_XZ_DEC_SPARC=y
CONFIG_XZ_DEC_BCJ=y
# CONFIG_XZ_DEC_TEST is not set
CONFIG_DECOMPRESS_GZIP=y
CONFIG_REED_SOLOMON=m
CONFIG_REED_SOLOMON_DEC16=y
CONFIG_TEXTSEARCH=y
CONFIG_TEXTSEARCH_KMP=m
CONFIG_TEXTSEARCH_BM=m
CONFIG_TEXTSEARCH_FSM=m
CONFIG_HAS_IOMEM=y
CONFIG_HAS_IOPORT=y
CONFIG_HAS_DMA=y
CONFIG_NLATTR=y
CONFIG_GENERIC_ATOMIC64=y
CONFIG_LRU_CACHE=m
CONFIG_AVERAGE=y
# CONFIG_CORDIC is not set
#
# Kernel hacking
#
CONFIG_PRINTK_TIME=y
CONFIG_DEFAULT_MESSAGE_LOGLEVEL=4
CONFIG_ENABLE_WARN_DEPRECATED=y
CONFIG_ENABLE_MUST_CHECK=y
CONFIG_FRAME_WARN=2048
CONFIG_MAGIC_SYSRQ=y
CONFIG_STRIP_ASM_SYMS=y
CONFIG_UNUSED_SYMBOLS=y
CONFIG_DEBUG_FS=y
# CONFIG_HEADERS_CHECK is not set
# CONFIG_DEBUG_SECTION_MISMATCH is not set
CONFIG_DEBUG_KERNEL=y
# CONFIG_DEBUG_SHIRQ is not set
# CONFIG_LOCKUP_DETECTOR is not set
# CONFIG_HARDLOCKUP_DETECTOR is not set
CONFIG_DETECT_HUNG_TASK=y
CONFIG_DEFAULT_HUNG_TASK_TIMEOUT=120
# CONFIG_BOOTPARAM_HUNG_TASK_PANIC is not set
CONFIG_BOOTPARAM_HUNG_TASK_PANIC_VALUE=0
CONFIG_SCHED_DEBUG=y
# CONFIG_SCHEDSTATS is not set
CONFIG_TIMER_STATS=y
# CONFIG_DEBUG_OBJECTS is not set
# CONFIG_SLUB_DEBUG_ON is not set
# CONFIG_SLUB_STATS is not set
# CONFIG_DEBUG_KMEMLEAK is not set
# CONFIG_DEBUG_RT_MUTEXES is not set
# CONFIG_RT_MUTEX_TESTER is not set
# CONFIG_DEBUG_SPINLOCK is not set
# CONFIG_DEBUG_MUTEXES is not set
# CONFIG_DEBUG_LOCK_ALLOC is not set
# CONFIG_PROVE_LOCKING is not set
# CONFIG_SPARSE_RCU_POINTER is not set
# CONFIG_LOCK_STAT is not set
# CONFIG_DEBUG_ATOMIC_SLEEP is not set
# CONFIG_DEBUG_LOCKING_API_SELFTESTS is not set
CONFIG_STACKTRACE=y
# CONFIG_DEBUG_STACK_USAGE is not set
# CONFIG_DEBUG_KOBJECT is not set
CONFIG_DEBUG_BUGVERBOSE=y
CONFIG_DEBUG_INFO=y
# CONFIG_DEBUG_INFO_REDUCED is not set
# CONFIG_DEBUG_VM is not set
# CONFIG_DEBUG_WRITECOUNT is not set
# CONFIG_DEBUG_MEMORY_INIT is not set
# CONFIG_DEBUG_LIST is not set
# CONFIG_TEST_LIST_SORT is not set
# CONFIG_DEBUG_SG is not set
# CONFIG_DEBUG_NOTIFIERS is not set
# CONFIG_DEBUG_CREDENTIALS is not set
# CONFIG_RCU_TORTURE_TEST is not set
# CONFIG_KPROBES_SANITY_TEST is not set
# CONFIG_BACKTRACE_SELF_TEST is not set
# CONFIG_DEBUG_BLOCK_EXT_DEVT is not set
# CONFIG_DEBUG_FORCE_WEAK_PER_CPU is not set
# CONFIG_LKDTM is not set
# CONFIG_FAULT_INJECTION is not set
# CONFIG_LATENCYTOP is not set
CONFIG_SYSCTL_SYSCALL_CHECK=y
# CONFIG_DEBUG_PAGEALLOC is not set
CONFIG_NOP_TRACER=y
```

```
CONFIG_HAVE_FUNCTION_TRACER=y
CONFIG_HAVE_FUNCTION_GRAPH_TRACER=y
CONFIG_HAVE_DYNAMIC_FTRACE=y
CONFIG_HAVE_FTRACE_MCOUNT_RECORD=y
CONFIG_HAVE_SYSCALL_TRACEPOINTS=y
CONFIG_RING_BUFFER=y
CONFIG_EVENT_TRACING=y
CONFIG_EVENT_POWER_TRACING_DEPRECATED=y
CONFIG_CONTEXT_SWITCH_TRACER=y
CONFIG_RING_BUFFER_ALLOW_SWAP=y
CONFIG_TRACING=y
CONFIG_GENERIC_TRACER=y
CONFIG_TRACING_SUPPORT=y
CONFIG_FTRACE=y
# CONFIG_FUNCTION_TRACER is not set
# CONFIG_IRQSOFF_TRACER is not set
# CONFIG_SCHED_TRACER is not set
# CONFIG_FTRACE_SYSCALLS is not set
CONFIG_BRANCH_PROFILE_NONE=y
# CONFIG_PROFILE_ANNOTATED_BRANCHES is not set
# CONFIG_PROFILE_ALL_BRANCHES is not set
# CONFIG_STACK_TRACER is not set
CONFIG_BLK_DEV_IO_TRACE=y
CONFIG_KPROBE_EVENT=y
# CONFIG_FTRACE_STARTUP_TEST is not set
# CONFIG_RING_BUFFER_BENCHMARK is not set
# CONFIG_DYNAMIC_DEBUG is not set
# CONFIG_DMA_API_DEBUG is not set
# CONFIG_ATOMIC64_SELFTEST is not set
# CONFIG_ASYNC_RAID6_TEST is not set
# CONFIG_SAMPLES is not set
CONFIG_HAVE_ARCH_KGDB=y
# CONFIG_KGDB is not set
# CONFIG_TEST_KSTRTOX is not set
# CONFIG_PPC_DISABLE_WERROR is not set
CONFIG_PPC_WERROR=y
CONFIG_PRINT_STACK_DEPTH=64
# CONFIG_DEBUG_STACKOVERFLOW is not set
# CONFIG_PPC_EMULATED_STATS is not set
# CONFIG_CODE_PATCHING_SELFTEST is not set
# CONFIG_FTR_FIXUP_SELFTEST is not set
# CONFIG_MSI_BITMAP_SELFTEST is not set
# CONFIG_XMON is not set
# CONFIG_VIRQ_DEBUG is not set
# CONFIG_BDI_SWITCH is not set
# CONFIG_PPC_EARLY_DEBUG is not set
#
# Security options
#
CONFIG_KEYS=y
# CONFIG_TRUSTED_KEYS is not set
# CONFIG_ENCRYPTED_KEYS is not set
CONFIG_KEYS_DEBUG_PROC_KEYS=y
# CONFIG_SECURITY_DMESG_RESTRICT is not set
CONFIG_SECURITY=y
CONFIG_SECURITYFS=y
CONFIG_SECURITY_NETWORK=y
CONFIG_SECURITY_NETWORK_XFRM=y
CONFIG_SECURITY_PATH=y
CONFIG_LSM_MMAP_MIN_ADDR=65536
CONFIG_SECURITY_SELINUX=y
CONFIG_SECURITY_SELINUX_BOOTPARAM=y
CONFIG_SECURITY_SELINUX_BOOTPARAM_VALUE=0
CONFIG_SECURITY_SELINUX_DISABLE=y
CONFIG_SECURITY_SELINUX_DEVELOP=y
CONFIG_SECURITY_SELINUX_AVC_STATS=y
CONFIG_SECURITY_SELINUX_CHECKREQPROT_VALUE=1
# CONFIG_SECURITY_SELINUX_POLICYDB_VERSION_MAX is not set
CONFIG_SECURITY_TOMOYO=y
CONFIG_SECURITY_TOMOYO_MAX_ACCEPT_ENTRY=2048
CONFIG_SECURITY_TOMOYO_MAX_AUDIT_LOG=1024
# CONFIG_SECURITY_TOMOYO_OMIT_USERSPACE_LOADER is not set
CONFIG_SECURITY_TOMOYO_POLICY_LOADER="/sbin/tomoyo-init"
CONFIG_SECURITY_TOMOYO_ACTIVATION_TRIGGER="/sbin/init"
# CONFIG_SECURITY_APPARMOR is not set
# CONFIG_IMA is not set
# CONFIG_EVM is not set
CONFIG_DEFAULT_SECURITY_SELINUX=y
# CONFIG_DEFAULT_SECURITY_TOMOYO is not set
# CONFIG_DEFAULT_SECURITY_DAC is not set
CONFIG_DEFAULT_SECURITY="selinux"
CONFIG_XOR_BLOCKS=m
CONFIG_ASYNC_CORE=m
CONFIG_ASYNC_MEMCPY=m
CONFIG_ASYNC_XOR=m
CONFIG_ASYNC_PQ=m
CONFIG_ASYNC_RAID6_RECOV=m
CONFIG_CRYPTO=y
#
# Crypto core or helper
#
```

```
CONFIG_CRYPTO_ALGAPI=y
CONFIG_CRYPTO_ALGAPI2=y
CONFIG_CRYPTO_AEAD=m
CONFIG_CRYPTO_AEAD2=y
CONFIG_CRYPTO_BLKCIPHER=m
CONFIG_CRYPTO_BLKCIPHER2=y
CONFIG_CRYPTO_HASH=y
CONFIG_CRYPTO_HASH2=y
CONFIG_CRYPTO_RNG=m
CONFIG_CRYPTO_RNG2=y
CONFIG_CRYPTO_PCOMP=m
CONFIG_CRYPTO_PCOMP2=y
CONFIG_CRYPTO_MANAGER=y
CONFIG_CRYPTO_MANAGER2=y
# CONFIG_CRYPTO_USER is not set
CONFIG_CRYPTO_MANAGER_DISABLE_TESTS=y
CONFIG_CRYPTO_GF128MUL=m
CONFIG_CRYPTO_NULL=m
CONFIG_CRYPTO_WORKQUEUE=y
# CONFIG_CRYPTO_CRYPTD is not set
CONFIG_CRYPTO_AUTHENC=m
CONFIG_CRYPTO_TEST=m
#
# Authenticated Encryption with Associated Data
#
CONFIG_CRYPTO_CCM=m
CONFIG_CRYPTO_GCM=m
CONFIG_CRYPTO_SEQIV=m
#
# Block modes
#
CONFIG_CRYPTO_CBC=m
CONFIG_CRYPTO_CTR=m
CONFIG_CRYPTO_CTS=m
CONFIG_CRYPTO_ECB=m
CONFIG_CRYPTO_LRW=m
CONFIG_CRYPTO_PCBC=m
CONFIG_CRYPTO_XTS=m
#
# Hash modes
#
CONFIG_CRYPTO_HMAC=m
CONFIG_CRYPTO_XCBC=m
CONFIG_CRYPTO_VMAC=m
#
# Digest
#
CONFIG_CRYPTO_CRC32C=m
CONFIG_CRYPTO_GHASH=m
CONFIG_CRYPTO_MD4=m
CONFIG_CRYPTO_MD5=y
CONFIG_CRYPTO_MICHAEL_MIC=m
CONFIG_CRYPTO_RMD128=m
CONFIG_CRYPTO_RMD160=m
CONFIG_CRYPTO_RMD256=m
CONFIG_CRYPTO_RMD320=m
CONFIG_CRYPTO_SHA1=m
CONFIG_CRYPTO_SHA256=m
CONFIG_CRYPTO_SHA512=m
CONFIG_CRYPTO_TGR192=m
CONFIG_CRYPTO_WP512=m
#
# Ciphers
#
CONFIG_CRYPTO_AES=m
CONFIG_CRYPTO_ANUBIS=m
CONFIG_CRYPTO_ARC4=m
CONFIG_CRYPTO_BLOWFISH=m
CONFIG_CRYPTO_BLOWFISH_COMMON=m
CONFIG_CRYPTO_CAMELLIA=m
CONFIG_CRYPTO_CAST5=m
CONFIG_CRYPTO_CAST6=m
CONFIG_CRYPTO_DES=m
CONFIG_CRYPTO_FCRYPT=m
CONFIG_CRYPTO_KHAZAD=m
CONFIG_CRYPTO_SALSA20=m
CONFIG_CRYPTO_SEED=m
CONFIG_CRYPTO_SERPENT=m
CONFIG_CRYPTO_TEA=m
CONFIG_CRYPTO_TWOFISH=m
CONFIG_CRYPTO_TWOFISH_COMMON=m
#
# Compression
#
CONFIG_CRYPTO_DEFLATE=m
CONFIG_CRYPTO_ZLIB=m
CONFIG_CRYPTO_LZO=m
#
# Random Number Generation
#
CONFIG_CRYPTO_ANSI_CPRNG=m
```

```
# CONFIG_CRYPTO_USER_API_HASH is not set
# CONFIG_CRYPTO_USER_API_SKCIPHER is not set
CONFIG_CRYPTO_HW=y
# CONFIG_CRYPTO_DEV_PPC4XX is not set
# CONFIG_PPC_CLOCK is not set
# CONFIG_VIRTUALIZATION is not set
```

Configuración Buildroot (Archivo .config)

```
# Automatically generated make config: don't edit
# Buildroot 2013.02 Configuration
#
BR2_HAVE_DOT_CONFIG=y
BR2_HOSTARCH_NEEDS_IA32_LIBS=y
# BR2_arm is not set
# BR2_armeb is not set
# BR2_aarch64 is not set
# BR2_avr32 is not set
# BR2_bfin is not set
# BR2_i386 is not set
# BR2_microblazeel is not set
# BR2_microblazebe is not set
# BR2_mips is not set
# BR2_mipsel is not set
# BR2_mips64 is not set
# BR2_mips64el is not set
BR2_powerpc=y
# BR2_sh is not set
# BR2_sh64 is not set
# BR2_sparc is not set
# BR2_x86_64 is not set
# BR2_xtensa is not set
BR2_ARCH="powerpc"
BR2_ENDIAN="BIG"
BR2_GCC_TARGET_TUNE="405"
# BR2_generic_powerpc is not set
# BR2_powerpc_401 is not set
# BR2_powerpc_403 is not set
BR2_powerpc_405=y
# BR2_powerpc_405fp is not set
# BR2_powerpc_440 is not set
# BR2_powerpc_440fp is not set
# BR2_powerpc_464 is not set
# BR2_powerpc_464fp is not set
# BR2_powerpc_476 is not set
# BR2_powerpc_476fp is not set
# BR2_powerpc_505 is not set
# BR2_powerpc_601 is not set
# BR2_powerpc_602 is not set
# BR2_powerpc_603 is not set
# BR2_powerpc_603e is not set
# BR2_powerpc_604 is not set
# BR2_powerpc_604e is not set
# BR2_powerpc_620 is not set
# BR2_powerpc_630 is not set
# BR2_powerpc_740 is not set
# BR2_powerpc_7400 is not set
# BR2_powerpc_7450 is not set
# BR2_powerpc_750 is not set
# BR2_powerpc_821 is not set
# BR2_powerpc_823 is not set
# BR2_powerpc_860 is not set
# BR2_powerpc_970 is not set
# BR2_powerpc_8540 is not set
# BR2_powerpc_8548 is not set
# BR2_powerpc_e300c2 is not set
# BR2_powerpc_e300c3 is not set
# BR2_powerpc_e500mc is not set
BR2_powerpc_CLASSIC=y
#
# Build options
#
#
# Commands
#
BR2_WGET="wget --passive-ftp -nd -t 3"
BR2_SVN="svn"
BR2_BZR="bzr"
BR2_GIT="git"
BR2_LOCALFILES="cp"
BR2_SCP="scp"
BR2_SSH="ssh"
BR2_HG="hg"
BR2_ZCAT="gzip -d -c"
BR2_BZCAT="bzcat"
BR2_XZCAT="xzcat"
BR2_TAR_OPTIONS=""
BR2_DEFCONFIG="$(CONFIG_DIR)/defconfig"
BR2_DL_DIR="$(HOME)/sources"
```

```
BR2_HOST_DIR="$(BASE_DIR)/host"
#
# Mirrors and Download locations
#
BR2_PRIMARY_SITE=""
BR2_BACKUP_SITE="http://sources.buildroot.net/"
BR2_KERNEL_MIRROR="http://www.kernel.org/pub/"
BR2_GNU_MIRROR="http://ftp.gnu.org/pub/gnu"
BR2_DEBIAN_MIRROR="http://ftp.debian.org"
BR2_JLEVEL=0
# BR2_CCACHE is not set
# BR2_DEPRECATED is not set
# BR2_ENABLE_DEBUG is not set
BR2_STRIP_strip=y
# BR2_STRIP_sstrip is not set
# BR2_STRIP_none is not set
BR2_STRIP_EXCLUDE_FILES=""
BR2_STRIP_EXCLUDE_DIRS=""
# BR2_OPTIMIZE_0 is not set
# BR2_OPTIMIZE_1 is not set
# BR2_OPTIMIZE_2 is not set
# BR2_OPTIMIZE_3 is not set
BR2_OPTIMIZE_S=y
# BR2_PREFER_STATIC_LIB is not set
BR2_PACKAGE_OVERRIDE_FILE="$(TOPDIR)/local.mk"
#
# Toolchain
#
# BR2_TOOLCHAIN_BUILDROOT is not set
BR2_TOOLCHAIN_EXTERNAL=y
# BR2_TOOLCHAIN_CTNG is not set
# BR2_TOOLCHAIN_EXTERNAL_CODESOURCERY_POWERPC201103 is not set
BR2_TOOLCHAIN_EXTERNAL_CODESOURCERY_POWERPC201009=y
# BR2_TOOLCHAIN_EXTERNAL_CUSTOM is not set
BR2_TOOLCHAIN_EXTERNAL_DOWNLOAD=y
# BR2_TOOLCHAIN_EXTERNAL_PREINSTALLED is not set
BR2_TOOLCHAIN_EXTERNAL_PREFIX="powerpc-linux-gnu"
BR2_TOOLCHAIN_EXTERNAL_GLIBC=y
#
# Gdb Options
#
# BR2_PACKAGE_GDB is not set
# BR2_TOOLCHAIN_EXTERNAL_GDB_SERVER_COPY is not set
BR2_LARGEFILE=y
BR2_INET_IPV6=y
BR2_TOOLCHAIN_HAS_NATIVE_RPC=y
BR2_USE_WCHAR=y
BR2_ENABLE_LOCALE=y
BR2_INSTALL_LIBSTDCPP=y
BR2_TOOLCHAIN_HAS_THREADS=y
BR2_TOOLCHAIN_HAS_THREADS_DEBUG=y
BR2_TOOLCHAIN_HAS_THREADS_DEBUG_IF_NEEDED=y
BR2_TOOLCHAIN_HAS_SHADOW_PASSWORDS=y
# BR2_ENABLE_LOCALE_PURGE is not set
BR2_GENERATE_LOCALE=""
BR2_USE_MMU=y
BR2_TARGET_OPTIMIZATION="-pipe"
BR2_TARGET_LDFLAGS=""
# BR2_ECLIPSE_REGISTER is not set
#
# System configuration
#
BR2_TARGET_GENERIC_HOSTNAME="xupv2p-antares"
BR2_TARGET_GENERIC_ISSUE="Welcome to antares"
# BR2_TARGET_GENERIC_PASSWD_DES is not set
BR2_TARGET_GENERIC_PASSWD_MD5=y
# BR2_TARGET_GENERIC_PASSWD_SHA256 is not set
# BR2_TARGET_GENERIC_PASSWD_SHA512 is not set
BR2_TARGET_GENERIC_PASSWD_METHOD="md5"
BR2_ROOTFS_DEVICE_CREATION_STATIC=y
# BR2_ROOTFS_DEVICE_CREATION_DYNAMIC_DEVTMPFS is not set
# BR2_ROOTFS_DEVICE_CREATION_DYNAMIC_MDEV is not set
# BR2_ROOTFS_DEVICE_CREATION_DYNAMIC_UDEV is not set
BR2_INIT_BUSYBOX=y
# BR2_INIT_SYSV is not set
#
# systemd requires largefile, wchar, IPv6, threads and udev support
#
# BR2_INIT_NONE is not set
BR2_ROOTFS_DEVICE_TABLE="system/device_table.txt"
BR2_ROOTFS_STATIC_DEVICE_TABLE="system/device_table_dev.txt"
BR2_ROOTFS_SKELETON_DEFAULT=y
# BR2_ROOTFS_SKELETON_CUSTOM is not set
BR2_TARGET_GENERIC_ROOT_PASSWD="antares"
BR2_TARGET_GENERIC_GETTY_PORT="ttyUL0"
# BR2_TARGET_GENERIC_GETTY_BAUDRATE_KEEP is not set
# BR2_TARGET_GENERIC_GETTY_BAUDRATE_9600 is not set
# BR2_TARGET_GENERIC_GETTY_BAUDRATE_19200 is not set
# BR2_TARGET_GENERIC_GETTY_BAUDRATE_38400 is not set
# BR2_TARGET_GENERIC_GETTY_BAUDRATE_57600 is not set
BR2_TARGET_GENERIC_GETTY_BAUDRATE_115200=y
```

```
BR2_TARGET_GENERIC_GETTY_BAUDRATE="115200"
BR2_TARGET_GENERIC_GETTY_TERM="vt100"
# BR2_TARGET_GENERIC_REMOUNT_ROOTFS_RW is not set
BR2_ROOTFS_OVERLAY=""
BR2_ROOTFS_POST_BUILD_SCRIPT=""
BR2_ROOTFS_POST_IMAGE_SCRIPT=""
#
# Package Selection for the target
#
BR2_PACKAGE_BUSYBOX=y
# BR2_BUSYBOX_VERSION_1_19_X is not set
BR2_BUSYBOX_VERSION_1_20_X=y
# BR2_BUSYBOX_VERSION_1_21_X is not set
# BR2_PACKAGE_BUSYBOX_SNAPSHOT is not set
BR2_BUSYBOX_VERSION="1.20.2"
BR2_PACKAGE_BUSYBOX_CONFIG="package/busybox/busybox-1.20.x.config"
BR2_PACKAGE_BUSYBOX_SHOW_OTHERS=y
BR2_PACKAGE_BUSYBOX_WATCHDOG=y
BR2_PACKAGE_BUSYBOX_WATCHDOG_PERIOD="5"
#
# Audio and video applications
#
# BR2_PACKAGE_ALSA_UTILS is not set
# BR2_PACKAGE_AUMIX is not set
# BR2_PACKAGE_BELLAGIO is not set
# BR2_PACKAGE_FAAD2 is not set
# BR2_PACKAGE_FLAC is not set
# BR2_PACKAGE_FFMPEG is not set
# BR2_PACKAGE_GSTREAMER is not set
# BR2_PACKAGE_LAME is not set
# BR2_PACKAGE_MADPLAY is not set
# BR2_PACKAGE_MPD is not set
# BR2_PACKAGE_MPG123 is not set
# BR2_PACKAGE_MPLAYER is not set
# BR2_PACKAGE_MUSEPACK is not set
# BR2_PACKAGE_OPUS_TOOLS is not set
# BR2_PACKAGE_PULSEAUDIO is not set
# BR2_PACKAGE_VORBIS_TOOLS is not set
# BR2_PACKAGE_WAVPACK is not set
# BR2_PACKAGE_YAVTA is not set
#
# Compressors and decompressors
#
BR2_PACKAGE_BZIP2=y
BR2_PACKAGE_GZIP=y
# BR2_PACKAGE_INFOZIP is not set
# BR2_PACKAGE_LZOP is not set
# BR2_PACKAGE_XZ is not set
#
# Debugging, profiling and benchmark
#
# BR2_PACKAGE_BONNIE is not set
# BR2_PACKAGE_CACHE_CALIBRATOR is not set
# BR2_PACKAGE_DHRYSTONE is not set
# BR2_PACKAGE_DSTAT is not set
# BR2_PACKAGE_DMALLOC is not set
# BR2_PACKAGE_KEXEC is not set
# BR2_PACKAGE_LATENCYTOP is not set
# BR2_PACKAGE_LMBENCH is not set
# BR2_PACKAGE_LSOF is not set
# BR2_PACKAGE_LTP_TESTSUITE is not set
# BR2_PACKAGE_LTRACE is not set
# BR2_PACKAGE_MEMSTAT is not set
# BR2_PACKAGE_NETPERF is not set
# BR2_PACKAGE_OPROFILE is not set
#
# perf only available if Linux kernel is enabled, and requires largefile support
#
# BR2_PACKAGE_RAMSPEED is not set
# BR2_PACKAGE_RT_TESTS is not set
# BR2_PACKAGE_STRACE is not set
BR2_PACKAGE_STRESS=y
# BR2_PACKAGE_SYSPROF is not set
# BR2_PACKAGE_WHETSTONE is not set
# BR2_PACKAGE_VALGRIND is not set
# BR2_PACKAGE_PV is not set
#
# Development tools
#
# BR2_PACKAGE_BINUTILS is not set
# BR2_PACKAGE_BISON is not set
# BR2_PACKAGE_BSDIFF is not set
# BR2_PACKAGE_COREUTILS is not set
# BR2_PACKAGE_CVS is not set
# BR2_PACKAGE_DIFFUTILS is not set
# BR2_PACKAGE_DISTCC is not set
# BR2_PACKAGE_FINDUTILS is not set
# BR2_PACKAGE_FLEX is not set
# BR2_PACKAGE_GAWK is not set
# BR2_PACKAGE_GMP is not set
# BR2_PACKAGE_GPERF is not set
```

```
# BR2_PACKAGE_GREP is not set
# BR2_PACKAGE_MPC is not set
# BR2_PACKAGE_MPFR is not set
# BR2_PACKAGE_LIBTOOL is not set
# BR2_PACKAGE_M4 is not set
# BR2_PACKAGE_PATCH is not set
# BR2_PACKAGE_PKGCONF is not set
# BR2_PACKAGE_SED is not set
# BR2_PACKAGE_SSTRIP is not set
BR2_PACKAGE_TAR=y
# BR2_PACKAGE_VALA is not set
#
# Games
#
# BR2_PACKAGE_GNUCHESS is not set
# BR2_PACKAGE_PRBOOM is not set
#
# Graphic libraries and applications (graphic/text)
#
#
# Graphic applications
#
# BR2_PACKAGE_RRDTOOL is not set
#
# graphic libraries
#
# BR2_PACKAGE_CEGUI06 is not set
# BR2_PACKAGE_DIRECTFB is not set
# BR2_PACKAGE_FBDUMP is not set
# BR2_PACKAGE_FBGRAB is not set
# BR2_PACKAGE_FBSET is not set
# BR2_PACKAGE_FBTERM is not set
# BR2_PACKAGE_FBV is not set
# BR2_PACKAGE_FB_TEST_APP is not set
# BR2_PACKAGE_IMAGEMAGICK is not set
# BR2_PACKAGE_SDL is not set
#
# other GUIs
#
# BR2_PACKAGE_EFL is not set
# BR2_PACKAGE_QT is not set
# BR2_PACKAGE_XORG7 is not set
#
# X libraries and helper libraries
#
# BR2_PACKAGE_LIBERATION is not set
#
# X Window managers
#
#
# X applications
#
# BR2_PACKAGE_GOB2 is not set
#
# midori requires C++, WCHAR in toolchain and libgtk2
#
#
# Filesystem and flash utilities
#
# BR2_PACKAGE_CIFS_UTILS is not set
# BR2_PACKAGE_CRAMFS is not set
# BR2_PACKAGE_CURLFTPFS is not set
# BR2_PACKAGE_DOSFSTOOLS is not set
# BR2_PACKAGE_E2FSPROGS is not set
# BR2_PACKAGE_FLASHBENCH is not set
# BR2_PACKAGE_GENEXT2FS is not set
# BR2_PACKAGE_GENROMFS is not set
# BR2_PACKAGE_MAKEDEVS is not set
# BR2_PACKAGE_MTD is not set
# BR2_PACKAGE_NFS_UTILS is not set
# BR2_PACKAGE_NTFS_3G is not set
# BR2_PACKAGE_SQUASHFS is not set
# BR2_PACKAGE_SSHFS is not set
# BR2_PACKAGE_UNIONFS is not set
# BR2_PACKAGE_XFSPROGS is not set
#
# Hardware handling
#
#
# Misc devices firmwares
#
# BR2_PACKAGE_B43_FIRMWARE is not set
# BR2_PACKAGE_LINUX_FIRMWARE is not set
# BR2_PACKAGE_UX500_FIRMWARE is not set
# BR2_PACKAGE_ZD1211_FIRMWARE is not set
# BR2_PACKAGE_CDRKIT is not set
# BR2_PACKAGE_DBUS is not set
# BR2_PACKAGE_DEVMEM2 is not set
# BR2_PACKAGE_DMRAID is not set
# BR2_PACKAGE_DVB_APPS is not set
# BR2_PACKAGE_DVBSNOOP is not set
```

```
# BR2_PACKAGE_EEPROG is not set
# BR2_PACKAGE_EVTEST is not set
# BR2_PACKAGE_FCONFIG is not set
# BR2_PACKAGE_FIS is not set
# BR2_PACKAGE_FMTOOLS is not set
# BR2_PACKAGE_FXLOAD is not set
# BR2_PACKAGE_GADGETFS_TEST is not set
# BR2_PACKAGE_GDISK is not set
# BR2_PACKAGE_GPSD is not set
# BR2_PACKAGE_GVFS is not set
# BR2_PACKAGE_HDPARM is not set
# BR2_PACKAGE_HWDATA is not set
# BR2_PACKAGE_I2C_TOOLS is not set
# BR2_PACKAGE_INPUT_EVENT_DAEMON is not set
# BR2_PACKAGE_INPUT_TOOLS is not set
# BR2_PACKAGE_IOSTAT is not set
# BR2_PACKAGE_IRDA_UTILS is not set
# BR2_PACKAGE_KBD is not set
# BR2_PACKAGE_LCDPROC is not set
# BR2_PACKAGE_LM_SENSORS is not set
# BR2_PACKAGE_LSHW is not set
# BR2_PACKAGE_LSUIO is not set
# BR2_PACKAGE_LVM2 is not set
# BR2_PACKAGE_MDADM is not set
# BR2_PACKAGE_MEDIA_CTL is not set
# BR2_PACKAGE_MEMTESTER is not set
# BR2_PACKAGE_MINICOM is not set
# BR2_PACKAGE_NANOCOM is not set
# BR2_PACKAGE_NEARD is not set
# BR2_PACKAGE_OFONO is not set
# BR2_PACKAGE_OPEN2300 is not set
# BR2_PACKAGE_OPENOCD is not set
# BR2_PACKAGE_PARTED is not set
# BR2_PACKAGE_PCIUTILS is not set
# BR2_PACKAGE_PICOCOM is not set
# BR2_PACKAGE_RNG_TOOLS is not set
# BR2_PACKAGE_SANE_BACKENDS is not set
# BR2_PACKAGE_SDPARM is not set
# BR2_PACKAGE_SETSERIAL is not set
# BR2_PACKAGE_SG3_UTILS is not set
# BR2_PACKAGE_SMARTMONTOOLS is not set
# BR2_PACKAGE_SNOWBALL_HDMISERVICE is not set
# BR2_PACKAGE_SREDIRD is not set
# BR2_PACKAGE_STATSERIAL is not set
# BR2_PACKAGE_SYSSTAT is not set
# BR2_PACKAGE_TI_UTILS is not set
# BR2_PACKAGE_UBOOT_TOOLS is not set
#
# udev requires /dev mgmnt set to udev under System configuration
#
#
# udisks requires /dev mgmnt set to udev under System configuration
#
# BR2_PACKAGE_USB_MODESWITCH is not set
# BR2_PACKAGE_USB_MODESWITCH_DATA is not set
# BR2_PACKAGE_USBUTILS is not set
# BR2_PACKAGE_WIPE is not set
#
# Interpreter languages and scripting
#
# BR2_PACKAGE_ERLANG is not set
# BR2_PACKAGE_HASERL is not set
# BR2_PACKAGE_JAMVM is not set
# BR2_PACKAGE_LUA is not set
# BR2_PACKAGE_LUAJIT is not set
BR2_PACKAGE_PERL=y
BR2_PACKAGE_PERL_MODULES=""
#
# Perl libraries/modules
#
# BR2_PACKAGE_PHP is not set
BR2_PACKAGE_PYTHON=y
# BR2_PACKAGE_PYTHON_PY_ONLY is not set
BR2_PACKAGE_PYTHON_PYC_ONLY=y
# BR2_PACKAGE_PYTHON_PY_PYC is not set
#
# core python modules
#
#
# The following modules are unusual or require extra libraries
#
BR2_PACKAGE_PYTHON_BZIP2=y
# BR2_PACKAGE_PYTHON_BSDDB is not set
# BR2_PACKAGE_PYTHON_CODECSCJK is not set
# BR2_PACKAGE_PYTHON_CURSES is not set
# BR2_PACKAGE_PYTHON_PYEXPAT is not set
# BR2_PACKAGE_PYTHON_READLINE is not set
# BR2_PACKAGE_PYTHON_SSL is not set
BR2_PACKAGE_PYTHON_UNICODEDATA=y
# BR2_PACKAGE_PYTHON_SQLITE is not set
BR2_PACKAGE_PYTHON_ZLIB=y
```

```
# BR2_PACKAGE_PYTHON3 is not set
#
# external python modules
#
# BR2_PACKAGE_PYTHON_BOTTLE is not set
# BR2_PACKAGE_PYTHON_DPKT is not set
# BR2_PACKAGE_PYTHON_ID3 is not set
# BR2_PACKAGE_PYTHON_MAD is not set
# BR2_PACKAGE_PYTHON_MELD3 is not set
# BR2_PACKAGE_PYTHON_NETIFACES is not set
# BR2_PACKAGE_PYTHON_NFC is not set
# BR2_PACKAGE_PYTHON_PROTOBUF is not set
# BR2_PACKAGE_PYTHON_PYGAME is not set
# BR2_PACKAGE_PYTHON_PYPARSING is not set
# BR2_PACKAGE_PYTHON_SERIAL is not set
# BR2_PACKAGE_PYTHON_SETUPTOOLS is not set
# BR2_PACKAGE_RUBY is not set
# BR2_PACKAGE_TCL is not set
#
# Libraries
#
#
# Audio/Sound
#
# BR2_PACKAGE_ALSA_LIB is not set
# BR2_PACKAGE_AUDIOFILE is not set
# BR2_PACKAGE_CELT051 is not set
# BR2_PACKAGE_LIBAO is not set
# BR2_PACKAGE_LIBCDAUDIO is not set
# BR2_PACKAGE_LIBCUE is not set
# BR2_PACKAGE_LIBCUEFILE is not set
# BR2_PACKAGE_LIBID3TAG is not set
# BR2_PACKAGE_LIBLO is not set
# BR2_PACKAGE_LIBMAD is not set
# BR2_PACKAGE_LIBMPD is not set
# BR2_PACKAGE_LIBREPLAYGAIN is not set
# BR2_PACKAGE_LIBSAMPLERATE is not set
# BR2_PACKAGE_LIBSNDFILE is not set
# BR2_PACKAGE_LIBVORBIS is not set
# BR2_PACKAGE_OPUS is not set
# BR2_PACKAGE_PORTAUDIO is not set
# BR2_PACKAGE_SPEEX is not set
# BR2_PACKAGE_TAGLIB is not set
# BR2_PACKAGE_TREMOR is not set
#
# Compression and decompression
#
# BR2_PACKAGE_LIBARCHIVE is not set
# BR2_PACKAGE_LZO is not set
BR2_PACKAGE_ZLIB=y
#
# Crypto
#
# BR2_PACKAGE_BEECRYPT is not set
# BR2_PACKAGE_GNUTLS is not set
# BR2_PACKAGE_LIBGCRYPT is not set
# BR2_PACKAGE_LIBGPG_ERROR is not set
# BR2_PACKAGE_LIBMCRYPT is not set
# BR2_PACKAGE_LIBMHASH is not set
# BR2_PACKAGE_LIBNSS is not set
# BR2_PACKAGE_LIBSHA1 is not set
# BR2_PACKAGE_NETTLE is not set
BR2_PACKAGE_OCF_LINUX=y
BR2_PACKAGE_OPENSSL=y
BR2_PACKAGE_OPENSSL_BIN=y
BR2_PACKAGE_OPENSSL_ENGINES=y
BR2_PACKAGE_OPENSSL_OCF=y
# BR2_PACKAGE_POLARSSL is not set
#
# Database
#
# BR2_PACKAGE_BERKELEYDB is not set
# BR2_PACKAGE_GDBM is not set
# BR2_PACKAGE_MYSQL_CLIENT is not set
# BR2_PACKAGE_SQLCIPHER is not set
# BR2_PACKAGE_SQLITE is not set
#
# Filesystem
#
# BR2_PACKAGE_GAMIN is not set
# BR2_PACKAGE_LIBCONFIG is not set
# BR2_PACKAGE_LIBCONFUSE is not set
# BR2_PACKAGE_LIBFUSE is not set
# BR2_PACKAGE_LIBLOCKFILE is not set
# BR2_PACKAGE_LIBSYSFS is not set
#
# Graphics
#
# BR2_PACKAGE_ATK is not set
# BR2_PACKAGE_CAIRO is not set
# BR2_PACKAGE_FONTCONFIG is not set
```

```
# BR2_PACKAGE_FREETYPE is not set
# BR2_PACKAGE_GD is not set
# BR2_PACKAGE_IMLIB2 is not set
# BR2_PACKAGE_JPEG is not set
# BR2_PACKAGE_LIBART is not set
# BR2_PACKAGE_LIBDMTX is not set
# BR2_PACKAGE_LIBEXIF is not set
# BR2_PACKAGE_LIBGEOTIFF is not set
# BR2_PACKAGE_GDK_PIXBUF is not set
# BR2_PACKAGE_LIBPNG is not set
# BR2_PACKAGE_LIBRAW is not set
# BR2_PACKAGE_LIBRSVG is not set
# BR2_PACKAGE_LIBSVGTINY is not set
# BR2_PACKAGE_LIBUNGIF is not set
# BR2_PACKAGE_OPENCV is not set
# BR2_PACKAGE_PANGO is not set
# BR2_PACKAGE_PIXMAN is not set
# BR2_PACKAGE_TIFF is not set
#
# webkit requires C++, WCHAR in toolchain and libgtk2
#
# BR2_PACKAGE_ZXING is not set
#
# Hardware handling
#
# BR2_PACKAGE_CCID is not set
# BR2_PACKAGE_LCDAPI is not set
# BR2_PACKAGE_LIBAIO is not set
#
# libatasmart requires udev to be enabled
#
# BR2_PACKAGE_LIBRAW1394 is not set
# BR2_PACKAGE_TSLIB is not set
# BR2_PACKAGE_LIBFREEFARE is not set
# BR2_PACKAGE_LIBFTDI is not set
# BR2_PACKAGE_LIBHID is not set
# BR2_PACKAGE_LIBIQRF is not set
# BR2_PACKAGE_LIBNFC is not set
# BR2_PACKAGE_LIBNFC_LLCP is not set
# BR2_PACKAGE_LIBUSB is not set
# BR2_PACKAGE_LIBV4L is not set
# BR2_PACKAGE_MTDEV is not set
# BR2_PACKAGE_NEARDAL is not set
# BR2_PACKAGE_PCSC_LITE is not set
#
# Javascript
#
# BR2_PACKAGE_EXPLORERCANVAS is not set
# BR2_PACKAGE_FLOT is not set
# BR2_PACKAGE_JQUERY is not set
# BR2_PACKAGE_JQUERY_SPARKLINE is not set
# BR2_PACKAGE_JQUERY_VALIDATION is not set
# BR2_PACKAGE_JSMIN is not set
#
# Multimedia
#
# BR2_PACKAGE_LIBDVDREAD is not set
# BR2_PACKAGE_LIBDVDNAV is not set
# BR2_PACKAGE_LIBEBML is not set
# BR2_PACKAGE_LIBMATROSKA is not set
# BR2_PACKAGE_LIBMMS is not set
# BR2_PACKAGE_LIBMPEG2 is not set
# BR2_PACKAGE_LIBOGG is not set
# BR2_PACKAGE_LIBPLAYER is not set
# BR2_PACKAGE_LIBTHEORA is not set
# BR2_PACKAGE_LIVE555 is not set
# BR2_PACKAGE_MEDIASTREAMER is not set
#
# Networking
#
BR2_PACKAGE_GLIB_NETWORKING=y
# BR2_PACKAGE_LIBCGI is not set
# BR2_PACKAGE_LIBCGICC is not set
# BR2_PACKAGE_LIBCURL is not set
# BR2_PACKAGE_LIBDNET is not set
# BR2_PACKAGE_LIBESMTP is not set
# BR2_PACKAGE_LIBEXOSIP2 is not set
# BR2_PACKAGE_LIBFCGI is not set
# BR2_PACKAGE_LIBGSASL is not set
# BR2_PACKAGE_LIBIDN is not set
# BR2_PACKAGE_LIBISCSI is not set
# BR2_PACKAGE_LIBOAUTH is not set
# BR2_PACKAGE_LIBMICROHTTPD is not set
# BR2_PACKAGE_NEON is not set
BR2_PACKAGE_LIBMNL=y
# BR2_PACKAGE_LIBMODBUS is not set
# BR2_PACKAGE_LIBMBUS is not set
# BR2_PACKAGE_LIBNETFILTER_ACCT is not set
BR2_PACKAGE_LIBNETFILTER_CONNTRACK=y
# BR2_PACKAGE_LIBNETFILTER_CTHELPER is not set
BR2_PACKAGE_LIBNETFILTER_CTTIMEOUT=y
```

```
# BR2_PACKAGE_LIBNETFILTER_LOG is not set
# BR2_PACKAGE_LIBNETFILTER_QUEUE is not set
BR2_PACKAGE_LIBNFNETLINK=y
# BR2_PACKAGE_LIBNL is not set
# BR2_PACKAGE_LIBOPING is not set
BR2_PACKAGE_LIBPCAP=y
# BR2_PACKAGE_LIBOSIP2 is not set
# BR2_PACKAGE_LIBRSYNC is not set
# BR2_PACKAGE_LIBSOUP is not set
# BR2_PACKAGE_LIBTIRPC is not set
# BR2_PACKAGE_LIBTORRENT is not set
# BR2_PACKAGE_LIBUPNP is not set
# BR2_PACKAGE_LIBVNCSERVER is not set
# BR2_PACKAGE_ORTP is not set
# BR2_PACKAGE_SLIRP is not set
# BR2_PACKAGE_USBREDIR is not set
# BR2_PACKAGE_ZEROMQ is not set
#
# Other
#
# BR2_PACKAGE_APR is not set
# BR2_PACKAGE_APR_UTIL is not set
# BR2_PACKAGE_ELFUTILS is not set
# BR2_PACKAGE_FFTW is not set
# BR2_PACKAGE_LIBARGTABLE2 is not set
# BR2_PACKAGE_ARGP_STANDALONE is not set
# BR2_PACKAGE_BOOST is not set
# BR2_PACKAGE_LIBATOMIC_OPS is not set
BR2_PACKAGE_LIBCAP=y
# BR2_PACKAGE_LIBCAP_NG is not set
# BR2_PACKAGE_LIBDAEMON is not set
# BR2_PACKAGE_LIBELF is not set
# BR2_PACKAGE_LIBEVENT is not set
# BR2_PACKAGE_LIBEV is not set
BR2_PACKAGE_LIBFFI=y
# BR2_PACKAGE_GSL is not set
BR2_PACKAGE_LIBGLIB2=y
# BR2_PACKAGE_LIBICAL is not set
# BR2_PACKAGE_LIBNSPR is not set
# BR2_PACKAGE_LIBSIGC is not set
# BR2_PACKAGE_LIBTPL is not set
# BR2_PACKAGE_LIBURCU is not set
# BR2_PACKAGE_LINUX_PAM is not set
# BR2_PACKAGE_LTTNG_LIBUST is not set
# BR2_PACKAGE_MTDEV2TUIO is not set
# BR2_PACKAGE_ORC is not set
# BR2_PACKAGE_POCO is not set
# BR2_PACKAGE_PROTOBUF is not set
# BR2_PACKAGE_SCHIFRA is not set
# BR2_PACKAGE_LIBLOG4C_LOCALTIME is not set
#
# Text and terminal handling
#
# BR2_PACKAGE_ENCHANT is not set
# BR2_PACKAGE_LIBFRIBIDI is not set
# BR2_PACKAGE_ICU is not set
# BR2_PACKAGE_LINENOISE is not set
BR2_PACKAGE_NCURSES=y
# BR2_PACKAGE_NCURSES_TARGET_PANEL is not set
# BR2_PACKAGE_NCURSES_TARGET_FORM is not set
# BR2_PACKAGE_NCURSES_TARGET_MENU is not set
# BR2_PACKAGE_NEWT is not set
# BR2_PACKAGE_PCRE is not set
# BR2_PACKAGE_POPT is not set
BR2_PACKAGE_READLINE=y
# BR2_PACKAGE_SLANG is not set
#
# JSON/XML
#
# BR2_PACKAGE_CJSON is not set
# BR2_PACKAGE_EXPAT is not set
# BR2_PACKAGE_EZXML is not set
# BR2_PACKAGE_JSON_C is not set
# BR2_PACKAGE_LIBROXML is not set
# BR2_PACKAGE_LIBXML2 is not set
# BR2_PACKAGE_LIBXSLT is not set
# BR2_PACKAGE_LIBYAML is not set
# BR2_PACKAGE_MXML is not set
# BR2_PACKAGE_XERCES is not set
# BR2_PACKAGE_YAJL is not set
#
# Miscellaneous
#
# BR2_PACKAGE_COLLECTD is not set
# BR2_PACKAGE_EMPTY is not set
# BR2_PACKAGE_GOOGLEFONTDIRECTORY is not set
# BR2_PACKAGE_MCRYPT is not set
# BR2_PACKAGE_MOBILE_BROADBAND_PROVIDER_INFO is not set
# BR2_PACKAGE_SHARED_MIME_INFO is not set
# BR2_PACKAGE_SNOWBALL_INIT is not set
# BR2_PACKAGE_SOUND_THEME_BOREALIS is not set
```

```
# BR2_PACKAGE_SOUND_THEME_FREEDESKTOP is not set
#
# Networking applications
#
# BR2_PACKAGE_ARGUS is not set
# BR2_PACKAGE_ARPTABLES is not set
# BR2_PACKAGE_AVAHI is not set
# BR2_PACKAGE_AXEL is not set
# BR2_PACKAGE_BLUEZ_UTILS is not set
# BR2_PACKAGE_BOA is not set
# BR2_PACKAGE_BIND is not set
# BR2_PACKAGE_BMON is not set
BR2_PACKAGE_BRIDGE_UTILS=y
# BR2_PACKAGE_CAN_UTILS is not set
# BR2_PACKAGE_CONNMAN is not set
# BR2_PACKAGE_CTORRENT is not set
# BR2_PACKAGE_CONNTRACK_TOOLS is not set
# BR2_PACKAGE_CUPS is not set
# BR2_PACKAGE_DHCP is not set
BR2_PACKAGE_DHCPDUMP=y
# BR2_PACKAGE_DNSMASQ is not set
# BR2_PACKAGE_DROPBEAR is not set
# BR2_PACKAGE_EBTABLES is not set
BR2_PACKAGE_ETHTOOL=y
# BR2_PACKAGE_GESFTPSERVER is not set
# BR2_PACKAGE_HEIRLOOM_MAILX is not set
# BR2_PACKAGE_HIAWATHA is not set
# BR2_PACKAGE_HOSTAPD is not set
# BR2_PACKAGE_HTTPING is not set
# BR2_PACKAGE_IFPLUGD is not set
# BR2_PACKAGE_IFTOP is not set
# BR2_PACKAGE_INADYN is not set
# BR2_PACKAGE_IPERF is not set
BR2_PACKAGE_IPROUTE2=y
# BR2_PACKAGE_IPSEC_TOOLS is not set
# BR2_PACKAGE_IPSET is not set
# BR2_PACKAGE_IPTABLES is not set
# BR2_PACKAGE_IW is not set
# BR2_PACKAGE_KISMET is not set
# BR2_PACKAGE_LIGHTTPD is not set
# BR2_PACKAGE_LINKS is not set
# BR2_PACKAGE_LINPHONE is not set
# BR2_PACKAGE_LRZSZ is not set
# BR2_PACKAGE_MACCHANGER is not set
# BR2_PACKAGE_MII_DIAG is not set
# BR2_PACKAGE_MROUTED is not set
# BR2_PACKAGE_MSMTP is not set
BR2_PACKAGE_MUTT=y
# BR2_PACKAGE_NBD is not set
# BR2_PACKAGE_NCFTP is not set
# BR2_PACKAGE_NDISC6 is not set
BR2_PACKAGE_NETCAT=y
# BR2_PACKAGE_NETATALK is not set
# BR2_PACKAGE_NETPLUG is not set
# BR2_PACKAGE_NETSNMP is not set
BR2_PACKAGE_NETSTAT_NAT=y
# BR2_PACKAGE_NFACCT is not set
# BR2_PACKAGE_NOIP is not set
# BR2_PACKAGE_NGIRCD is not set
# BR2_PACKAGE_NGREP is not set
# BR2_PACKAGE_NTP is not set
# BR2_PACKAGE_NUTTCP is not set
# BR2_PACKAGE_OLSR is not set
# BR2_PACKAGE_OPENNTPD is not set
BR2_PACKAGE_OPENSSH=y
# BR2_PACKAGE_OPENSWAN is not set
# BR2_PACKAGE_OPENVPN is not set
BR2_PACKAGE_PORTMAP=y
# BR2_PACKAGE_PPPD is not set
# BR2_PACKAGE_PPTP_LINUX is not set
# BR2_PACKAGE_PROFTPD is not set
# BR2_PACKAGE_PROXYCHAINS_NG is not set
# BR2_PACKAGE_QUAGGA is not set
# BR2_PACKAGE_RADVD is not set
# BR2_PACKAGE_RPCBIND is not set
# BR2_PACKAGE_RSH_REDONE is not set
# BR2_PACKAGE_RSYNC is not set
# BR2_PACKAGE_RTORRENT is not set
# BR2_PACKAGE_SAMBA is not set
# BR2_PACKAGE_SCONESERVER is not set
# BR2_PACKAGE_SER2NET is not set
# BR2_PACKAGE_SOCAT is not set
# BR2_PACKAGE_SOCKETCAND is not set
# BR2_PACKAGE_SPAWN_FCGI is not set
# BR2_PACKAGE_SPICE_PROTOCOL is not set
BR2_PACKAGE_SQUID=y
# BR2_PACKAGE_STUNNEL is not set
# BR2_PACKAGE_TCPDUMP is not set
# BR2_PACKAGE_TCPREPLAY is not set
# BR2_PACKAGE_TFTPD is not set
# BR2_PACKAGE_THTTPD is not set
```

```
# BR2_PACKAGE_TINYHTTPD is not set
# BR2_PACKAGE_TN5250 is not set
# BR2_PACKAGE_TRANSMISSION is not set
# BR2_PACKAGE_TVHEADEND is not set
# BR2_PACKAGE_UDPCAST is not set
# BR2_PACKAGE_ULOGD is not set
# BR2_PACKAGE_USHARE is not set
# BR2_PACKAGE_VDE2 is not set
# BR2_PACKAGE_VPNC is not set
# BR2_PACKAGE_VSFTPD is not set
# BR2_PACKAGE_VTUN is not set
BR2_PACKAGE_WGET=y
# BR2_PACKAGE_WIRELESS_TOOLS is not set
# BR2_PACKAGE_WPA_SUPPLICANT is not set
# BR2_PACKAGE_XINETD is not set
# BR2_PACKAGE_XL2TP is not set
#
# Package managers
#
# BR2_PACKAGE_IPKG is not set
# BR2_PACKAGE_OPKG is not set
#
# rpm requires libneon with SSL, XML and ZLIB support
#
#
# Real-Time
#
# BR2_PACKAGE_XENOMAI is not set
#
# Shell and utilities
#
# BR2_PACKAGE_AT is not set
BR2_PACKAGE_BASH=y
# BR2_PACKAGE_DASH is not set
# BR2_PACKAGE_DIALOG is not set
# BR2_PACKAGE_FILE is not set
# BR2_PACKAGE_GNUPG is not set
# BR2_PACKAGE_INOTIFY_TOOLS is not set
# BR2_PACKAGE_LOCKFILE_PROGS is not set
# BR2_PACKAGE_LOGROTATE is not set
# BR2_PACKAGE_LOGSURFER is not set
# BR2_PACKAGE_SCREEN is not set
# BR2_PACKAGE_SUDO is not set
# BR2_PACKAGE_TIME is not set
# BR2_PACKAGE_WHICH is not set
# BR2_PACKAGE_XMLSTARLET is not set
#
# System tools
#
# BR2_PACKAGE_ACL is not set
# BR2_PACKAGE_ATTR is not set
# BR2_PACKAGE_BOOTUTILS is not set
# BR2_PACKAGE_BWM_NG is not set
# BR2_PACKAGE_CPULOAD is not set
# BR2_PACKAGE_HTOP is not set
# BR2_PACKAGE_KEYUTILS is not set
# BR2_PACKAGE_KMOD is not set
# BR2_PACKAGE_MODULE_INIT_TOOLS is not set
# BR2_PACKAGE_MONIT is not set
# BR2_PACKAGE_NCDU is not set
# BR2_PACKAGE_POLKIT is not set
# BR2_PACKAGE_PROCPS is not set
# BR2_PACKAGE_PSMISC is not set
# BR2_PACKAGE_QUOTA is not set
# BR2_PACKAGE_RSYSLOG is not set
# BR2_PACKAGE_SYSKLOGD is not set
# BR2_PACKAGE_SYSVINIT is not set
# BR2_PACKAGE_SUPERVISOR is not set
#
# systemd not available (depends on /dev management with udev and ipv6 support, and thread support in
toolchain)
#
# BR2_PACKAGE_UTIL_LINUX is not set
#
# Text editors and viewers
#
# BR2_PACKAGE_ED is not set
# BR2_PACKAGE_LESS is not set
BR2_PACKAGE_NANO=y
BR2_PACKAGE_NANO_TINY=y
# BR2_PACKAGE_UEMACS is not set
BR2_PACKAGE_VIM=y
BR2_PACKAGE_VIM_RUNTIME=y
#
# Host utilities
#
# BR2_PACKAGE_HOST_DFU_UTIL is not set
# BR2_PACKAGE_HOST_LPC3250LOADER is not set
# BR2_PACKAGE_HOST_OPENOCD is not set
# BR2_PACKAGE_HOST_SAM_BA is not set
# BR2_PACKAGE_HOST_UBOOT_TOOLS is not set
```

```
#
# Filesystem images
#
# BR2_TARGET_ROOTFS_CLOOP is not set
# BR2_TARGET_ROOTFS_CPIO is not set
# BR2_TARGET_ROOTFS_CRAMFS is not set
BR2_TARGET_ROOTFS_EXT2=y
BR2_TARGET_ROOTFS_EXT2_BLOCKS=0
BR2_TARGET_ROOTFS_EXT2_INODES=0
BR2_TARGET_ROOTFS_EXT2_RESBLKS=0
# BR2_TARGET_ROOTFS_EXT2_NONE is not set
BR2_TARGET_ROOTFS_EXT2_GZIP=y
# BR2_TARGET_ROOTFS_EXT2_BZIP2 is not set
# BR2_TARGET_ROOTFS_EXT2_LZMA is not set
#
# initramfs requires a Linux kernel to be built
#
# BR2_TARGET_ROOTFS_JFFS2 is not set
# BR2_TARGET_ROOTFS_ROMFS is not set
# BR2_TARGET_ROOTFS_SQUASHFS is not set
BR2_TARGET_ROOTFS_TAR=y
# BR2_TARGET_ROOTFS_TAR_NONE is not set
BR2_TARGET_ROOTFS_TAR_GZIP=y
# BR2_TARGET_ROOTFS_TAR_BZIP2 is not set
# BR2_TARGET_ROOTFS_TAR_LZMA is not set
BR2_TARGET_ROOTFS_TAR_OPTIONS=""
# BR2_TARGET_ROOTFS_UBIFS is not set
#
# Bootloaders
#
# BR2_TARGET_BAREBOX is not set
# BR2_TARGET_UBOOT is not set
#
# Kernel
#
# BR2_LINUX_KERNEL is not set
#
# Legacy config options
#
# BR2_PACKAGE_CUSTOMIZE is not set
# BR2_PACKAGE_XSERVER_xorg is not set
# BR2_PACKAGE_XSERVER_tinyx is not set
# BR2_PACKAGE_PTHREAD_STUBS is not set
# BR2_PACKAGE_GETTEXT_STATIC is not set
# BR2_PACKAGE_LIBINTL is not set
# BR2_PACKAGE_INPUT_TOOLS_EVTEST is not set
```

Configuración OpenSSL (Archivo openssl.cnf)

```
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#
# This definition stops the following lines choking if HOME isn't
# defined.
HOME = .
RANDFILE = $ENV::HOME/.rnd
# Extra OBJECT IDENTIFIER info:
#oid_file = $ENV::HOME/.oid
oid_section = new_oids
# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)
[ new_oids ]
# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6
####################################################################
[ ca ]
default_ca = CA_default # The default ca section
####################################################################
[ CA_default ]
dir = ./demoCA # Where everything is kept
certs = $dir/certs # Where the issued certs are kept
crl_dir = $dir/crl # Where the issued crl are kept
database = $dir/index.txt # database index file.
#unique_subject = no # Set to 'no' to allow creation of
# several ctificates with same subject.
new_certs_dir = $dir/newcerts # default place for new certs.
certificate = $dir/cacert.pem # The CA certificate
serial = $dir/serial # The current serial number
crlnumber = $dir/crlnumber # the current crl number
# must be commented out to leave a V1 CRL
crl = $dir/crl.pem # The current CRL
private_key = $dir/private/cakey.pem# The private key
```

```
RANDFILE = $dir/private/.rand # private random number file
x509_extensions = usr_cert # The extentions to add to the cert
# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt = ca_default # Subject Name options
cert_opt = ca_default # Certificate field options
# Extension copying option: use with caution.
# copy_extensions = copy
# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions = crl_ext
default_days = 365 # how long to certify for
default_crl_days= 30 # how long before next CRL
default_md = sha1 # which md to use.
preserve = no # keep passed DN ordering
# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy = policy_match
# For the CA policy
[ policy_match ]
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName = optional
stateOrProvinceName = optional
localityName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
####################################################################
[ req ]
default_bits = 1024
default_keyfile = privkey.pem
distinguished_name = req_distinguished_name
attributes = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert
# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret
# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr
# req_extensions = v3_req # The extensions to add to a certificate request
[ req_distinguished_name ]
countryName = Country Name (2 letter code)
countryName_default = AU
countryName_min = 2
countryName_max = 2
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Some-State
localityName = Locality Name (eg, city)
0.organizationName = Organization Name (eg, company)
0.organizationName_default = Internet Widgits Pty Ltd
# we can do this but it is not needed normally :-)
#1.organizationName = Second Organization Name (eg, company)
#1.organizationName_default = World Wide Web Pty Ltd
organizationalUnitName = Organizational Unit Name (eg, section)
#organizationalUnitName_default =
commonName = Common Name (eg, YOUR name)
commonName_max = 64
emailAddress = Email Address
emailAddress_max = 64
# SET-ex3 = SET extension number 3
[ req_attributes ]
challengePassword = A challenge password
challengePassword_min = 4
challengePassword_max = 20
unstructuredName = An optional company name
[ usr_cert ]
# These extensions are added when 'ca' signs a request.
# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.
basicConstraints=CA:FALSE
# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.
```

```
# This is OK for an SSL server.
# nsCertType = server
# For an object signing certificate this would be used.
# nsCertType = objsign
# For normal client use this is typical
# nsCertType = client, email
# and for everything including object signing:
# nsCertType = client, email, objsign
# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment
# This will be displayed in Netscape's comment listbox.
nsComment = "OpenSSL Generated Certificate"
# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move
# Copy subject details
# issuerAltName=issuer:copy
#nsCaRevocationUrl = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName
[ v3_req ]
# Extensions to add to a certificate request
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
[ v3_ca ]
# Extensions for a typical CA
# PKIX recommendation.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid:always,issuer:always
# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true
# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign
# Some might want this also
# nsCertType = sslCA, emailCA
# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy
# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF
[ crl_ext ]
# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.
# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
[ proxy_cert_ext ]
# These extensions should be added when creating a proxy certificate
# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.
basicConstraints=CA:FALSE
# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.
# This is OK for an SSL server.
# nsCertType = server
# For an object signing certificate this would be used.
# nsCertType = objsign
# For normal client use this is typical
# nsCertType = client, email
# and for everything including object signing:
# nsCertType = client, email, objsign
# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment
# This will be displayed in Netscape's comment listbox.
nsComment = "OpenSSL Generated Certificate"
# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always
# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy
# An alternative to produce certificates that aren't
# deprecated according to PKIX.
# subjectAltName=email:move
```

```
# Copy subject details
# issuerAltName=issuer:copy
#nsCaRevocationUrl = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName
# This really needs to be in place for it to be a proxy certificate.
proxyCertInfo=critical,language:id-ppl-anyLanguage,pathlen:3,policy:foo
```

Servidor (Archivo common.h)

```
#ifndef _common_h
#define _common_h
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <netdb.h>
#include <fcntl.h>
#include <signal.h>
#include <unistd.h>
#include <string.h>
#include <time.h>
#include <sys/stat.h>
#include <openssl/ssl.h>
#define CA_LIST "root.pem"
#define HOST "localhost"
#define RANDOM "random.pem"
#define PORT 4433
#define BUFSIZZ 1024
#define SERVER "ANTARES https server/1.0"
#define PROTOCOL "HTTP/1.1"
#define RFC1123FMT "%a, %d %b %Y %H:%M:%S GMT"
extern BIO *bio_err;
int berr_exit (char *string);
int err_exit(char *string);
SSL_CTX *initialize_ctx(char *keyfile, char *password);
void destroy_ctx(SSL_CTX *ctx);
#ifndef ALLOW_OLD_VERSIONS
#if (OPENSSL_VERSION_NUMBER < 0x00905100L)
#error "Must use OpenSSL 0.9.6 or later"
#endif
#endif
#endif
```

Servidor (Archivo common.c)

```
#include "common.h"
#include <openssl/err.h>
BIO *bio_err=0;
static char *pass;
static int password_cb(char *buf,int num,
int rwflag,void *userdata);
static void sigpipe_handle(int x);
/* Si se produce un error se sale de la rutina*/
int err_exit(string)
char *string;
{
fprintf(stderr,"%s\n",string);
exit(0);
}
/* Imprimir errores de SSL y salir*/
int berr_exit(string)
char *string;
{
BIO_printf(bio_err,"%s\n",string);
ERR_print_errors(bio_err);
exit(0);
}
/*Revisar si el password es seguro*/
static int password_cb(char *buf,int num,
int rwflag,void *userdata)
{
if(num<strlen(pass)+1)
return(0);
strcpy(buf,pass);
return(strlen(pass));
}
static void sigpipe_handle(int x){
}
SSL_CTX *initialize_ctx(keyfile,password)
char *keyfile;
char *password;
{
SSL_METHOD *meth;
SSL_CTX *ctx;
```

```
if(!bio_err){
/* Sistema de inicialización global*/
SSL_library_init();
SSL_load_error_strings();
bio_err=BIO_new_fp(stderr,BIO_NOCLOSE);
}
/* Crear un SIGPIPE handler para el caso de que el cliente mande
TCP RST en lugar de un close_notify*/
signal(SIGPIPE,sigpipe_handle);
/* Crear nuestro contexto*/
meth=SSLv23_method();
ctx=SSL_CTX_new(meth);
/* Leer nuestras claves y certificados*/
if(!(SSL_CTX_use_certificate_chain_file(ctx,
keyfile)))
berr_exit("Can't read certificate file");
pass=password;
SSL_CTX_set_default_passwd_cb(ctx,
password_cb);
if(!(SSL_CTX_use_PrivateKey_file(ctx,
keyfile,SSL_FILETYPE_PEM)))
berr_exit("Can't read key file");
/* Leer nuestas CA de confianza*/
if(!(SSL_CTX_load_verify_locations(ctx,
CA_LIST,0)))
berr_exit("Can't read CA list");
#if (OPENSSL_VERSION_NUMBER < 0x00905100L)
SSL_CTX_set_verify_depth(ctx,1);
#endif
return ctx;
}
void destroy_ctx(ctx)
SSL_CTX *ctx;
{
SSL_CTX_free(ctx);
}
```

Servidor (Archivo server.h)

```
#ifndef _server_h
#define _server_h
#define KEYFILE "server.pem"
#define PASSWORD "antares"
#define DHFILE "dhparam.pem"
int tcp_listen(void);
void load_dh_params(SSL_CTX *ctx,char *file);
void generate_eph_rsa_key(SSL_CTX *ctx);
void send_error(BIO *io, char *status, char *title, char *extra, char *text, char *path);
void send_file(BIO *io, char *path, struct stat *statbuf);
void send_headers(BIO *io, char *status, char *title, char *extra, char *mime, int length, time_t date);
#endif
```

Servidor (Archivo server.c)

```
#include "common.h"
#include "server.h"
int tcp_listen()
{
int sock;
struct sockaddr_in sin;
int val=1;
if((sock=socket(AF_INET,SOCK_STREAM,0))<0) //SOCK_STREAM establece el protocolo TCP para co-
municación
err_exit("Couldn't make socket");
memset(&sin,0,sizeof(sin));
sin.sin_addr.s_addr=INADDR_ANY; //INADDR_ANY Permite conexion con cualquier cliente
sin.sin_family=AF_INET; //AF_INET Se establece para comunicación con clientes remotos y clientes dentro
del mismo ordenador del servidor
sin.sin_port=htons(PORT); //htons convierte el entero a formato estandard de red
setsockopt(sock,SOL_SOCKET,SO_REUSEADDR,
&val,sizeof(val));
if(bind(sock,(struct sockaddr *)&sin, //Se liga al programa con el fichero (socket) creado
sizeof(sin))<0)
berr_exit("Couldn't bind");
listen(sock,5); //Se aceptan hasta 5 clientes en cola
return(sock);
}
void load_dh_params(ctx,file)
SSL_CTX *ctx;
char *file;
{
DH *ret=0;
BIO *bio;
if ((bio=BIO_new_file(file,"r")) == NULL)
berr_exit("Couldn't open DH file");
ret=PEM_read_bio_DHparams(bio,NULL,NULL,
NULL);
BIO_free(bio);
if(SSL_CTX_set_tmp_dh(ctx,ret)<0)
berr_exit("Couldn't set DH parameters");
}
```

```
void generate_eph_rsa_key(ctx)
SSL_CTX *ctx;
{
RSA *rsa;
rsa=RSA_generate_key(512,RSA_F4,NULL,NULL);
if (!SSL_CTX_set_tmp_rsa(ctx,rsa))
berr_exit("Couldn't set RSA key");
RSA_free(rsa);
}
```

Servidor (Archivo wserver.c)

```
/* SERVIDOR HTTPS BASICO */
#include "common.h"
#include "server.h"
static int client_auth=0;
static int fork_child=1;
static char *ciphers=0;
#define CLIENT_AUTH_REQUEST 1
#define CLIENT_AUTH_REQUIRE 2
#define CLIENT_AUTH_REHANDSHAKE 3
static int s_server_session_id_context = 1;
static int s_server_auth_session_id_context = 2;
static int http_serve(ssl,s)
SSL *ssl;
int s;
{
char buf[BUFSIZZ];
int r,len;
BIO *io,*ssl_bio;
char *method;
char *path;
char *protocol;
struct stat statbuf;
char pathbuf[4096];
io=BIO_new(BIO_f_buffer()); //preparamos para usar BIO_* functions
ssl_bio=BIO_new(BIO_f_ssl());
BIO_set_ssl(ssl_bio,ssl,BIO_CLOSE);
BIO_push(io,ssl_bio);
////////////////////////RECIBIR CABECERA (PRIMERA LINEA)//////////////////////////////
r=BIO_gets(io,buf,BUFSIZZ-1);
printf(buf);
method = strtok(buf, " ");
path = strtok(NULL, " ");
protocol = strtok(NULL, "\r");
if (!method || !path || !protocol) return -1;
////////////////////ANALISIS DE SOLICITUD//////////////////////////
if (strcasecmp(method, "GET") != 0)
send_error(io, "501", "Not supported", NULL, "Method is not supported.", path);
else if (stat(path, &statbuf) < 0)
send_error(io, "404", "Not Found", NULL, "File not found.", path);
else if (S_ISDIR(statbuf.st_mode))
{
len = strlen(path);
if (len == 0 || path[len - 1] != '/')
{
snprintf(pathbuf, sizeof(pathbuf), "Location: %s/", path);
send_error(io, "302", "Found", pathbuf, "Directories must end with a slash.", path);
}
else
{
snprintf(pathbuf, sizeof(pathbuf), "%sindex.html", path);
send_file(io, pathbuf, &statbuf);
}
}
else
send_file(io, path, &statbuf);
////////////////LEER EL RESTO DE LA CABECERA////////////////////////
while(1){
r=BIO_gets(io,buf,BUFSIZZ-1);
printf(buf);
switch(SSL_get_error(ssl,r)){
case SSL_ERROR_NONE:
len=r;
break;
case SSL_ERROR_ZERO_RETURN:
goto shutdown;
break;
default:
berr_exit("SSL read problem");
}
/* Buscar la linea en blanco que señala
el final de la cabecera HTTP */
if(!strcmp(buf,"\r\n") ||
!strcmp(buf,"\n"))
break;
}
/* Se lleva acabo la negociacion si fue solicitada*/
if(client_auth==CLIENT_AUTH_REHANDSHAKE){
SSL_set_verify(ssl,SSL_VERIFY_PEER |
SSL_VERIFY_FAIL_IF_NO_PEER_CERT,0);
```

```
SSL_set_session_id_context(ssl,
(void *)&s_server_auth_session_id_context,
sizeof(s_server_auth_session_id_context));
if(SSL_renegotiate(ssl)<=0)
berr_exit("SSL renegotiation error");
if(SSL_do_handshake(ssl)<=0)
berr_exit("SSL renegotiation error");
ssl->state=SSL_ST_ACCEPT;
if(SSL_do_handshake(ssl)<=0)
berr_exit("SSL renegotiation error");
}
//////////////////////CERRAR CONEXION//////////////////////////
r=BIO_flush(io);
shutdown:
r=SSL_shutdown(ssl);
if(!r){
shutdown(s,1);
r=SSL_shutdown(ssl);
}
switch(r){
case 1:
break;
case 0:
case -1:
default:
berr_exit("\n");
}
SSL_free(ssl);
close(s);
return(0);
}
char *get_mime_type(char *name)
{
char *ext = strrchr(name, '.');
if (!ext) return NULL;
if (strcmp(ext, ".html") == 0 || strcmp(ext, ".htm") == 0) return "text/html";
if (strcmp(ext, ".jpg") == 0 || strcmp(ext, ".jpeg") == 0) return "image/jpeg";
if (strcmp(ext, ".gif") == 0) return "image/gif";
if (strcmp(ext, ".png") == 0) return "image/png";
if (strcmp(ext, ".css") == 0) return "text/css";
if (strcmp(ext, ".au") == 0) return "audio/basic";
if (strcmp(ext, ".wav") == 0) return "audio/wav";
if (strcmp(ext, ".avi") == 0) return "video/x-msvideo";
if (strcmp(ext, ".mpeg") == 0 || strcmp(ext, ".mpg") == 0) return "video/mpeg";
if (strcmp(ext, ".mp3") == 0) return "audio/mpeg";
return NULL;
}
void send_headers(BIO *io, char *status, char *title, char *extra, char *mime,
int length, time_t date)
{
char data[4096];
int n;
FILE *file = fopen("/root/servidor/header", "w");
time_t now;
char timebuf[128];
fprintf(file, "%s %s %s\r\n", PROTOCOL, status, title);
fprintf(file, "Server: %s\r\n", SERVER);
now = time(NULL);
strftime(timebuf, sizeof(timebuf), RFC1123FMT, gmtime(&now));
fprintf(file, "Date: %s\r\n", timebuf);
if (extra) fprintf(file, "%s\r\n", extra);
if (mime) fprintf(file, "Content-Type: %s\r\n", mime);
if (length >= 0) fprintf(file, "Content-Length: %d\r\n", length);
if (date != -1)
{
strftime(timebuf, sizeof(timebuf), RFC1123FMT, gmtime(&date));
fprintf(file, "Last-Modified: %s\r\n", timebuf);
}
fprintf(file, "Connection: close\r\n");
fprintf(file, "\r\n");
fclose(file);
FILE *file2 = fopen("/root/servidor/header", "r");
while ((n = fread(data, 1, sizeof(data), file2)) > 0) BIO_puts(io,data);
fclose(file2);
}
void send_error(BIO *io, char *status, char *title, char *extra, char *text, char *path)
{
char data[4096];
int n;
FILE *file = fopen("/root/servidor/error", "w");
send_headers(io, status, title, extra, get_mime_type(path), -1, -1);
fprintf(file, "<HTML><HEAD><TITLE>%s %s</TITLE></HEAD>\r\n", status, title);
fprintf(file, "<BODY><H4>%s %s</H4>\r\n", status, title);
fprintf(file, "%s\r\n", text);
fprintf(file, "</BODY></HTML>\r\n");
fclose(file);
FILE *file2 = fopen("/root/servidor/error", "r");
while ((n = fread(data, 1, sizeof(data), file2)) > 0) BIO_puts(io,data);
fclose(file2);
}
void send_file(BIO *io, char *path, struct stat *statbuf)
```

```
{
char data[4096];
int n;
FILE *file = fopen(path, "r");
if (!file)
send_error(io, "403", "Forbidden", NULL, "Access denied.", path);
else
{
int length = S_ISREG(statbuf->st_mode) ? statbuf->st_size : -1;
send_headers(io, "200", "OK", NULL, get_mime_type(path), length, statbuf->st_mtime);
while ((n = fread(data, 1, sizeof(data), file)) > 0) BIO_write(io,data,n);
fclose(file);
}
}
int main(argc,argv)
int argc;
char **argv;
{
int sock,s;
BIO *sbio;
SSL_CTX *ctx;
SSL *ssl;
int r;
pid_t pid;
extern char *optarg;
int c;
while((c=getopt(argc,argv,"cCxna:"))!=-1){
switch(c){
case 'c':
client_auth=CLIENT_AUTH_REQUEST;
break;
case 'C':
client_auth=CLIENT_AUTH_REQUIRE;
break;
case 'x':
client_auth=CLIENT_AUTH_REHANDSHAKE;
break;
case 'n':
fork_child=0;
break;
case 'a':
if(!(ciphers=strdup(optarg)))
err_exit("Out of memory");
break;
}
}
/* Construimos nuestro contexto SSL*/
ctx=initialize_ctx(KEYFILE,PASSWORD);
load_dh_params(ctx,DHFILE);
SSL_CTX_set_session_id_context(ctx,
(void*)&s_server_session_id_context,
sizeof s_server_session_id_context);
/* Establecemos nuestra lista de algoritmos de cifrado */
if(ciphers){
SSL_CTX_set_cipher_list(ctx,ciphers);
}
switch(client_auth){
case CLIENT_AUTH_REQUEST:
SSL_CTX_set_verify(ctx,SSL_VERIFY_PEER,0);
break;
case CLIENT_AUTH_REQUIRE:
SSL_CTX_set_verify(ctx,SSL_VERIFY_PEER |
SSL_VERIFY_FAIL_IF_NO_PEER_CERT,0);
break;
case CLIENT_AUTH_REHANDSHAKE:
break;
}
sock=tcp_listen();
printf("\n*****************SERVIDOR HTTPS ANTARES V 1.0*****************\n");
printf("Escuchando por puerto: %d ...\n",PORT);
while(1){
if((s=accept(sock,0,0))<0)
err_exit("Problem accepting");
if(fork_child && (pid=fork())){
close(s);
}
else {
sbio=BIO_new_socket(s,BIO_NOCLOSE); //Se crea un canal de comunicacion seguro
ssl=SSL_new(ctx);
SSL_set_bio(ssl,sbio,sbio);
if((r=SSL_accept(ssl)<=0)) //Se realiza el handshake
berr_exit("SSL accept error");
http_serve(ssl,s);
if(fork_child)
exit(0);
}
}
destroy_ctx(ctx);
exit(0);

}
```