L4/Fiasco/L4Linux Kickstart

Cheng Guanghui, Nicholas Mc Guire

Distributed & Embedded Systems Lab, Lanzhou University
December 18, 2006

Contents

Version	Author	Date	Comment
1.0	Cheng Guanghui, Nicholas McGuire	4 Oct 2006	initial release
1.1	Cheng Guanghui, Nicholas McGuire	17 Nov 2006	Support DOpE
2.0	Cheng Guanghui, Nicholas McGuire	18 Dec 2006	Fix Bugs

1. Introduction

Before going into the details of installing the L4/Fiasco Microkernel, we need to set up a desk-top Linux system as a basis. We will describe this for the current Ubuntu dapper (Ubuntu 6.06). But we have tested this document in Ubuntu Breezy (Ubuntu 5.10), Ubuntu Dapper (Ubuntu 6.06), Slackware 10.2 and Debian 4.0. Basically it should not be fundamentally different from other distributions. If you do install it on a different distribution and can point out the differences, we would be happy to here from you and add your informations.

As this kickstart is intended for newbie - some things might seem irritatingly simple - but we just want to make sure this is complete - for people that know Linux and kernel development on Linux desktops, just skip through the first part and make sure there is nothing new - for the others we hope this is detailed enough to get you up and running in less than 2 hours ! (excluding compile time ;)

- what is ubuntu?

  Ubuntu is a complete Linux distribution derived from the debian distribution - it is a Linux kernel and a full operating system, freely available with both community and professional support. Ubuntu is a new Linux distribution, based on Debian GNU/Linux. It differs from Debian in that there is a release every 6 months, and support is provided for 18 months after release for each version; in this way, Ubuntu aims to provide an up-to-date yet reasonably stable operating system for the average user through the sole use of Free Software. For development systems we currently don't recommend it due to the target availability being limited to 18 months (which is really not sufficient for most projects) - but as it is becoming popular and we also are using it for the 8th RT-Linux Workshop in Lanzhou University, China - we used it for this kickstart session.

- what is DROPS?
  A good way to get confused is to start looking for a drops package or a drops directory - DROPS the Dresden Realtime Operating System is a set of software packages - so the entire /home/L4 directory in this kickstart represents DROPS (with addition of the grub related files).

- what is L4?
  L4 is the name of a second-generation microkernel originally developed by Jochen Liedtke. However, nowadays it is also a specification for various implementations of the L4 API.

- what is Fiasco?

  Well this Fiasco is not a fiasco... one of the L4 implementations is Fiasco, which is being developed at TU Dresden. It is sometimes referred to as L4/Fiasco to emphasize this relationship. It is a variant of L4 specifically targeting hard real-time environments with resource partitioning demands, so compared to other L4 implementations the performance may be less in some cases, the essence of L4/Fiasco is predictability.

- what is L4Env?

  L4/Fiasco - in fact all L4 implementations - were developed as components - L4Env comprises a set of user-level services and libraries that make using L4/Fiasco easier and provide a basis for an operating-system built on top of L4. It includes the oskit the L4Linux and the l4 core sources (fiasco and many many servers).

- what is L4Linux?

  As L4/Fiasco is real-time capable, it also is the foundation of the Dresden Real-Time Operating System (DROPS). It supports parallel execution of real-time and time-sharing applications. The latter also include L4Linux, a para-virtualized version of Linux, which is able to run unmodified Linux programs. L4Linux is a Linux kernel version that is implemented as a seperate architecture `arch/l4/` in the Linux kernel tree and that instead of talking directly to the hardware interacts with the L4/Fiasco microkernel via IPC. Thus L4Linux never talks directly to the hardware but calls into L4/Fiasco to interact with hardware - this has clear implications for device driver usage - so don't expect to insmod eepro200.ko to get access to your eepro card !

- what is L4dope?
  DOpE means "Desktop Operating Environment" and it aims to provide a graphical user interface for the DROPS. Because DROPS targets hard real time OS the author considered the real-time capability when he designed DOpE. Although the X-windows architecture is very powerful and popular it is too complex to modify it for the real-time requirement. In the end they adopt the strategy like the Tcl/Tk and they provide a platform independent command interface. This command interface is reponsible to generate the graphical interface elements. L4dope is very small and it is only less than 2M.

2. Preparing for building

This section covers the basic setup of the host Linux system on which you will install L4/Fiasco/L4Linux next. If you already have an installed Linux host with all necessary tools to compile a kernel, then you can skip this section (except for the GRUB related patching !).

2.1. Linux distribution

We have tried two different Linux distribution Slackware 10.2 and Ubuntu 5.1/6.06 for L4Linux. In common Slackware is a good choice for building L4/Fiasco and L4Linux because Slackware is a Linux distribution for development users. If you are newbie for L4Linux or L4/Fiasco you don't need to worried about missing packages on Slackware very much (which the exception of packages related to building the documentation). But in ubuntu the installation is painful and when compiling there are quite a few missing packages or commands that are not found. Luckily adding or removing packages in ubuntu is very very easy - that is one of the things it shares with debian.

2.2. Ubuntu 5.1/6.06: List of Packages and Patches

Be careful. In this part I listed many many packages which are necessary for builing L4Linux and L4/Fiasco in the host. But if your ubuntu is not new and you have used for a short time maybe you have already installed some packages. However, in order to avoid some errors you need to check out these packages in you ubuntu host.

- openssh-client
- openssh-server
- gcc-3.4
- g++-3.4
- libncurses5-dev
- libncurses5
- doxygen
- tk-brief
- latex2html

- gawk
- hyperlatex
- imagemagick
- zlib1g-dev
- bison
- byacc
- flex
- autoconf2.13
- automake1.9
- fig2ps
- transfig
- fig2sty
- fig2sxd
- nasm
- make
- cpp
- patch

Notes: openssh packages is necessary for every linux distribution unless you really want to live an unsafe life. We must use gcc to compile the L4Linux (In fact it is a linux kernel modified version). The L4/Fiasco is implemented in c++ so we must use g++. Pay attention the gcc and g++ version and in the Fiasco website they suggest the users to use the gcc or g++ 3.x. For this manual we choose the version 3.4 on ubuntu and 3.3.6 on Slackware (included by default). libncurses5 and libncurses5-dev is used to handle terminal menu interfaces and is needed for the default configuration tools.

Automake and autoconf are tools for generarting Makefile and configure script. Bison and byacc are GNU Project parser generator. Flex is fast lexical analyzer generator. zlib1g-dev is necessary for zlib. nasm is portable 80x86 assembler. imagemagick - Image manipulation programs. doxygen is generating document for various language. Others such as tk-brief, latex2html, hyperlatex, fig2ps,transfig,fig2sty and fig2sxd are necessary

for compiling the latex document in the whole process. If you are not familiar with some packages function. you could use "apt-cache search xxx — grep xxx" to search it or use "man" command, too.

2.3. Packages and patches on target

To create this directory strucuture download the following CVS trees into /home/L4.

```
root@rtl14:/~# mkdir /home/L4
root@rtl14:/~# cvs -d :pserver:guest@os.inf.tu-dresden.de:/home/remote-cvs login
password:guest
```

You could download these file like this:

```
root@rtl14:/~# cvs -z3 -d :pserver:guest@os.inf.tu-dresden.de:/home/remote-cvs
checkout oskit
root@rtl14:/~# cvs -z3 -d :pserver:guest@os.inf.tu-dresden.de:/home/remote-cvs
checkout oskit10
root@rtl14:/~# cvs -z3 -d :pserver:guest@os.inf.tu-dresden.de:/home/remote-cvs
checkout l4
root@rtl14:/~# cvs -z3 -d :pserver:guest@os.inf.tu-dresden.de:/home/remote-cvs
checkout l4linux-2.6
```

for more details and posible updates please check out the web page "http://os.inf.tu-dresden.de/drops/download.html"

As for grub-0.97,it can be downloaded from the official GNU download site with the command: wget ftp://alpha.gnu.org/gnu/grub/grub-0.97.tar.gz

As l4 needs some extended features in the grub boot-loader we need to apply a patch - this patch grub-0.97-os.1.diff can be download from adam's personal web page (http://os.inf.tu-dresden.de/~adam/) i.e. via wget `http://os.inf.tu-dresden.de/~adam/grub/0.97/grub-0.97-os.1.diff.gz`. This patch will add symbol expansion to grub, allowing to configure a lot of the settins in a simpler way, and is needed for using grub as initial file provider to boot the l4 system.

After the dowload the directory structure should be like this - if your directory structure is different pleas adjust the commands listed in this manual appropriately (or for simplicity simply move files until you have it as shown here).

- Microkernel related files/directories:

```
oskit/
oskit10/
l4/
l4linux-2.6/
```

- Bootloder related files/directories:

```
grub-0.97
grub-0.97-os.1.diff
```

3. Building the L4Linux/Fiasco

3.1. Install some packages in the host

3.1.1. In Ubuntu 5.10/6.02 or Debian

I assume your ubuntu is new installation you haven't modified the system. we start from the scratch.

- configure Ubuntu network

```
root@rtl14:/~# ifconfig eth0 xxx.xxx.xxx.xxx
root@rtl14:/~# route add default gw yyy.yyy.yyy.yyy eth0
```

- modified the ubuntu package source list : add a avaiable "apt" source packages

```
root@rtl14:/~# vi /etc/apt/sources.list
 deb ftp://mirror.lzu.edu.cn/ubuntu/ubuntu/ dapper main multiverse
restricted universe
 deb ftp://mirror.lzu.edu.cn/ubuntu/ubuntu/ dapper-backports main
multiverse restricted universe
 deb ftp://mirror.lzu.edu.cn/ubuntu/ubuntu/ dapper-proposed main
multiverse restricted universe
 deb ftp://mirror.lzu.edu.cn/ubuntu/ubuntu/ dapper-security main
multiverse restricted universe
 deb ftp://mirror.lzu.edu.cn/ubuntu/ubuntu/ dapper-updates main
multiverse restricted universe
 deb ftp://mirror.lzu.edu.cn/ubuntu/ubuntu-cn/ dapper main
multiverse restricted universe
```

Note: that you of course need to pick some mirror site that is resonably neer to your location - if you use the one from Lanzhou University in Europe you will not be too happy :)

- **upgrade the system and update the package list in the cache is simple as ubuntu has the debian apt tools as its basic package management.**

```
root@rtl14:/~# apt-get upgrade
root@rtl14:/~# apt-get update
```

- Install all the necessary packages listed in part 2 of II like this:

```
root@rtl14:/~# apt-get install xxx
```

NOTE: pay much attention to version of gcc and g++ !

- add the symbolic link of gcc and g++ command
  first find the location of gcc-3.4 and g++ 3.4

```
root@rtl14:/~# which gcc-3.4
/usr/bin/gcc-3.4
root@rtl14:/~# which g++-3.4
/usr/bin/g++-3.4
```

(on ubuntu dapper the two commands are in /usr/bin/) and create the symbolic link

```
root@rtl14:/~# cd /usr/bin/
root@rtl14:/usr/bin# ln -s gcc-3.4 gcc
root@rtl14:/usr/bin# ln -s g++-3.4 g++
```

3.1.2. For Slackware 10.2

This is a short section - we found no missing packages for Slackware 10.2 except for some latex2html related files needed to build the documentation - unfortunately we did not fix this problem yet - other than this all packages on slackware 10.2 seem to be sufficient.

Now that the host is ready we can go on to the next step and build L4/Fiasco and L4Linux.

3.2. Configure basic L4 environment

If the directory l4/tool/gcc-wrap exists please remove it. It is obsolete and nothing helpful.

```
root@rtl14:/# cd /home/L4/l4/tool/
root@rtl14:/# rm -fr gcc-wrap
```

First we need to create the build directory which will be used to store all the generated files, if you forget to pass the build directory (or to create it) you should get the message:

```
root@rtl14:/home/L4# make -C l4 config
make: Entering directory ‘/home/L4/l4’
mk/Makeconf:128: *** need to give builddir with O=.../builddir.  Stop.
make: Leaving directory ‘/home/L4/l4’
```

So first we create the needed build directory - it will contain the binaries created by the build process - and then we start with configuration.

```
root@rtl14:/home/L4# mkdir build
root@rtl14:/home/L4# make -C l4 config O=/home/L4/build
```

The target architecture consists of three parts - the actual architecture (x86 in the below case) the subarchitecture (586 below) and the L4-API version, V2 is the stable release - the rest more or less is experimental (or not yet quite comleted) - note that these are not Fiasco "home-brew" API - but are L4 sub-specifications that are fully specified (and in fact implemented in different variateions - see the l4linux.org for more information on other L4 implementations)

```
Target Architecture  --->
   (x86) Target Architecture
      CPU type: "586"
   (L4.V2) ABI
```

This is the top level directory in which you unpacked the archive from the proceedings CD or in which the CVS download placed the directories `l4, l4linux-2.6, oskit, oskit10`.

```
Paths and Directories  --->
   DROPS_STDDIR: "/home/L4/"
   DROPS_INSTDIR: "$(DROPS_STDDIR)"
```

Especially when reporting bugs to the mailing list pleas turn this on - its noisy and produced a few megabyte of data if logged to a file - but it really can help pin-point problems related to the build process. Pay attention to this directory DROPS_STDDIR which includes these directories such as l4, l4linux-2.6, oskit and oskit10.

```
Verboseness and Messages  --->
   [ ] Verbose dependency building
   [*] Verbose compilation and building
   [*] Short messages for compilation
   [*] Colored build-steps

Compilers and Tools  --->
```

Check the settings here - they should be ok by default but make sure they fit your system or strange things will happen. In case of errors during the build process pleas include the tools information.

```
Advanced  --->
(uclibc) Libc to use
[*] use ldso for dynamic linking
[*] Build shared libloader.s.so
[*] use system-call entry code in the KIP (absolutely)
[ ] Enable RELEASE flag
```

Save the configuration file and exit. If you have compiled the Linux kernel you will be familiar with this interface.

3.3. Configure the L4/Fiasco kernel

```
root@rtl14:/home/L4# cd /home/L4/l4/
```

The kernel itself is only a small part of the l4 tree (what else to expect from a microkernel) - so we first configure this core part of you system, it is easy to cause big problems if this part is not properly configured - so be carefull - and record your settings for later trace (or posts to the L4 mailing list!).

```
root@rtl14:/home/L4/l4# cd kernel/fiasco/
root@rtl14:/home/L4/l4/kernel/fiasco/# make menuconfig
```

```
[Y]      Prompt for experimental features
-->      Target System Options
   (PC)     Target Platform
   (IA-32)  Target CPU Family
   (i586)   Target processor
   [Y]      Compile with regparm=3
   [ ]      Enables workaround for AMD FPU security leak
```

if you are unsure you can use "cat /proc/cpuinfo" to check you processor type, note that the Fiasco UX (the user mode version) is hidden under Target Platform - though it will most likely also be running on a PC.

```
-->      Kernel Options
   (L4V2)   Kernel ABI Version
   -->      ABI Extensions
      [Y]      Deceit bit disables switch
      [Y]      Exception IPC (EXPERIMENTAL)
      [Y]      Handle and preserve segments (EXPERIMENTAL)
      [ ]      PL0 Hack (EXPERIMENTAL)
   (PIT)    Scheduling Timer
```

Using the PIT (Programmable Interrupt Timer) for scheduling is a safe bet, once this works, and if you system has an APIC (Advanced Programable Interrupt Controller) try it - generally performance will be better with the APIC.

```
   [ ]      Synchronize KIP time with time-stamp counter
   [ ]      Fine-grained CPU time
   [ ]      Enable I/O port protection

-->      Kernel Debugging
```

For the first shot you most likely want to leave most of this off, if you run into ANY problems during your work with L4/Fiasco this is the first place to turn on all of its internal debugging capabilities - delivering output from the builtin debug capabilities it will not be hard to get help from the L4 developers -

We will leave it at the defaults for this first loop

```
-->      Compiling and Building
```

Here you can set your compiler (if using a cross-compiler or something else than the system default gcc/g++

Once you are done input "x" to save and exit.

3.4. Compile the L4/Fiasco

Before we have configured the l4 well now we could start to compile all packages.

```
root@rtl14:/home/L4/l4/kernel/fiasco/# cd /home/L4/
root@rtl14:/home/L4/# make -C l4 O=/home/L4/build
```

This will take some time - we recommend at least 2 cups of coffee at this point or you could have a sleep and when you wake up it is ok.

Note that on Slackware 10.2 there is a known problem with building the documents so you probably want to turn the documents off (or ignore the document related errors). You will get the folowing error message on Slackeare 10.2:

```
! LaTeX Error: File ‘html.sty’ not found.

Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: sty)

Enter file name:
```

to continue just type in X (capital X) every time it ask for the you to "Enter file name" and compilation will then complete (with an error regarding docs but ok otherwise).

The build process now builds not only the fiasco microkernel, but also all the essential servers that will be needed to actually utilize fiasco, in a microkernel most of the resources are managed by external entities running in user-space as "process" or as microkernel people call it "servers", you can think of these a bit like unix-daemons. In the L4/Fiasco package there are a very large number of such servers provided, you don't need all of them to operate L4/Fiasco, many of them are example servers to help you get started. After the build completed the binaries are in the directory that was passed with the O=/PATH to make - in our case /home/L4/build. In this directory you can find many binary modules for DROPS.many binary modules for DROPS.

We will now describe in brevity the most important servers found in `build/bin/x86_586/l4v2`, note that fiasco it self shows up as "main" not fiasco - this is most likely only to confuse newbie ;)

- bootstrap:
  Well this is just what the name says - the bootstrap for L4. This module must be able to directly speak to basic devices like crt and serial lines, it includes decompression

code (gzip) and minimum facilities to get the system working (loader for elf, exec) it is responsible for initial setup of regions and the kernel page. Not too surprising this module is relatively architecture dependant.

- fiasco (main):
  The RT-capable L4 kernel, this really is the core of the L4/Fiasco system. This is the actual microkernel that enforces the temporal and spatial resource seperation at runtime (partitioning). It also contains excessive debugging capabilities built in to debug and monitor the system at runtime.

- sigma0:
  Sigma0 is the pagefaulthandler for L4, it is the minimum pager of the system (domains running under L4 can have there privat pager if they like). At system startup all pages are owned by sigma0.

- roottask:
  The roottask is a basic resource manager replacing the older RMGR (ResourcManaGeR), it is responsible for the following resources

  - physical memory
  - interrupts
  - tasks
  - small address spaces
  - launch of basic servers (i.e. loader)

  Note that you might still find some "RMGR" in error messages - which then most likely refer to services provided by the roottask or by l4env.

- events:
  This is a communication server to provide communication between tasks running under L4/Fiasco control. The module is a channel like system allowing for synchronous or asynchronous one-to-one, many-to-one, one-to-many and many-to-many schemes. Every task must register at the events server to receive or send messages.

- names:
  Basic namespace management, names implements the following functionality:

  - (un)register a name with a thread_id
  - unregister all names registered for a certain task
  - resolve thread_id to name and vice versa
  - query all registered thread_id's

  - ensures uniqunes of names (strings)

  basically all modules depend on names in some way.

- log:
  Basic console and loging support, it ensures message order and buffering (if necessary i.e. for telnet client).

- dm_phys:
  This is the physical dataspace manager for L4, it allows to assign different physical address ranges to different pools. Each memory pool exclusively manages an area of the physical memory and can be used to reserve and use certain memory areas for special purposes (i.e. ISA or DMA). The configuration of dmphys is static at runtime, once the memory pools are setup at system initialization they can't be modified any more - this allows to configure partitioning of memory.

- simple_ts:
  L4's simple task manager, basically this provides:

  - minimum task init (unique ID, stack, etc.)
  - task management functions (create, terminate, etc.)
  - a task queue
  - timeout management and management functions

- rtc:
  Teal Time Clock server, providing:

  - reading the RTC
  - calibrating the delay loop

  The RTC is abstracted in a way that servers accessing the RTC (i.e. L4Linux) no longer need any access to the actual RTC but call the rtc server only.

- l4io:
  l4io is a multiserver providing the omeag0 (interrupt manager) and basic I/O management. l4io provides the generic I/O manager interface as defined in the generic_io package. l4io is responsible for basic PCI maintenance (taken from the Linux kernel) further more it is responsihle for I/O ports, I/O memory, and ISA DMA channels. The interrupt management is handled in omega0 which is part of the l4io module. This module also contains a few odd-balls that were not worthy of a independant module (i.e. mtrr, jiffies thread etc.).

- l4exec:
  L4exec is the binary Interpreter for L4, its main responsibility is to interprets L4 binaries and creates the dataspaces which contain the programs sections. It also is responsible to resolve dependencies with library modules (which must be loaded ahead of any call to l4exec for a executable).

- bmodfs:
  Minimum boot time file provider, as it builds on mapping data sapces to file names it can work together with GRUB (grub loading all the modules to well defined addresses) and thus allows accessing the different modules by name. Basically this is a minimum flat FS providing address to name resolution.

- loader:
  The dynamic L4 Loader is able to load binaries at runtime. It depends on the L4 exec server as its binary interpreter (currently supporting ELF only). The loaders responsibility is to fetch files via the file provider (i.e. bmodfs at boot time) and hand them to the binary interpreter (currently exec).

3.5. Configure and Compile of L4Linux

3.5.1. Boot L4Linux in L4dope

```
root@rtl14:/home/L4/l4/kernel/fiasco/#cd /home/L4/l4linux-2.6/
root@rtl14:/home/L4/l4linux-2.6/# make clean
root@rtl14:/home/L4/l4linux-2.6/# make menuconfig
  L4Linux configuration  --->
     (/home/L4/build/) L4 tree build directory
        Target architecture (i386/x86 architecture)  --->
        [ ] Run L4Linux in userland only (for Fiasco-UX)
        (32) Kernel default memory size
     [*] Use tamed mode
        L4 build options --->
           (586) DROPS subarchitecture for i386
           [ ] Use sysenter versions of L4 system call binding
           [ ] Use system call entry code in the KIP (absolute)
           [*] Use ldso for loading
     Stub drivers  --->
           [*] Use the rtc server
           [ ] Block driver for the generic_blk interface
           [*] Framebuffer driver for l4con and DOpE (input/output)
           [*]   Support for the X Window System driver
```

```
        [ ] Network driver for oshkosh.
        [ ] Network driver for ORe
        [ ] Pseudo serial driver for console
    Debugging options --->
        [*] Debugging options
        [ ] Register Linux program names in the kernel debugger
        [ ] Show page faults and exceptions of Linux server
        [ ] Print error information on user program sefaults
        [ ] Do some statistics (NEW)
        [ ] IRQ spinning wheels in VGA mode (NEW)
        [ ]Count interrupt disables (NEW)

The left is about Linux-2.6 kernel configuration.
    Processor type and features  --->
        Processor family (Athlon/Duron/K7)  --->
    Device Drivers  --->
        Input device support  --->
                --- Generic input layer (needed for keyboard, mouse, ...)
                ---   Userland interfaces
                < >   Mouse interface
                < >   Joystick interface
                < >   Touchscreen interface
                < >   Event interface
                < >   Event debugging
                ---   Input Device Drivers
                      [ ]   Keyboards  --->
                      [ ]   Mouse  --->
                      [ ]   Joysticks  --->
                      [ ]   Touchscreens  --->
                      [ ]   Miscellaneous devices  --->
                      Hardware I/O ports  --->
                            < > Serial I/O support
                            < > Gameport support
        Character devices  --->
            [*] Virtual terminal
        Graphics support
            <*> Support for frame buffer devices
                Console display driver support  --->
                      [ ] VGA text console
                      < > MDA text console (dual-headed) (EXPERIMENTAL)
                      <*> Framebuffer Console support
```

```
                        [*]   Framebuffer Console Rotation
                        [ ] Select compiled-in fonts
        In other options you could keep the default.
```

Note: In the Linux configuration there are important rules to boot the L4Linux and L4dope successfully.

```
        Depency:L4Linux configuration  --->
               Stub drivers  --->
                      [*] Framebuffer driver for l4con and DOpE (input/output)
                      [*]   Support for the X Window System driver
        is dependent from
        Character devices  --->
               [*] Virtual terminal
        And
        Graphics support
        <*> Support for frame buffer devices
               Console display driver support  --->
                     <*> Framebuffer Console support
                     [*]   Framebuffer Console Rotation
        Disable:
              Disable the mouse and keyboard
              Disable VGA Console
              Disalbe Serial I/O support
       Biscally you clean all the options in (Input device support) and it is ok


root@rtl14:/home/L4/l4linux-2.6/# make
```

3.5.2. Boot L4Linux by remote serial console

```
root@rtl14:/home/L4/l4/kernel/fiasco/#cd /home/L4/l4linux-2.6/
root@rtl14:/home/L4/l4linux-2.6/# make clean
root@rtl14:/home/L4/l4linux-2.6/# make menuconfig
  L4Linux configuration  --->
     (/home/L4/build/) L4 tree build directory
        Target architecture (i386/x86 architecture)  --->
        [ ] Run L4Linux in userland only (for Fiasco-UX)
        (32) Kernel default memory size
     [*] Use tamed mode
```

```
        L4 build options --->
           (586) DROPS subarchitecture for i386
           [ ] Use sysenter versions of L4 system call binding
           [ ] Use system call entry code in the KIP (absolute)
           [*] Use ldso for loading
        Debugging options --->
           [ ] Register Linux program names in the kernel debugger
           [ ] Show page faults and exceptions of Linux server
           [ ] Print error information on user program sefaults
           [ ] Do some statistics (NEW)
           [ ] IRQ spinning wheels in VGA mode (NEW)
           [ ]Count interrupt disables (NEW)
       Stub drivers  --->
           [*] Use the rtc server
           [ ] Block driver for the generic_blk interface
           [ ] Framebuffer driver for l4con and DOpE (input/output)
           [ ]   Support for the X Window System driver
           [ ] Network driver for oshkosh.
           [ ] Network driver for ORe
           [ ] Pseudo serial driver for console
       Debugging options --->
           [*] Debugging options
           [ ] Register Linux program names in the kernel debugger
           [ ] Show page faults and exceptions of Linux server
           [ ] Print error information on user program sefaults
           [ ] Do some statistics (NEW)
           [ ] IRQ spinning wheels in VGA mode (NEW)
           [ ]Count interrupt disables (NEW)

The left is about Linux-2.6 kernel configuration.
Processor type and features  --->
          Processor family (Athlon/Duron/K7)  --->
Device Drivers  --->
   Character devices  --->
Serial drivers  --->
   <*> 8250/16550 and compatible serial support
                  [*]   Console on 8250/16550 and compatible serial port
                  <*>   8250/16550 PCI device support (NEW)
                    (4)   Maximum number of 8250/16550 serial ports (NEW)
                    (4)   Number of 8250/16550 serial ports to register at runtime (NEW)
```

3.6. Reinstall and Configure the GRUB

3.6.1. Reinstall the GRUB

```
root@rtl14:/home/L4/l4/l4linux-2.6/#cd  drops_home/grub-0.97
root@rtl14:/home/L4/grub-0.97/# patch -p0 <../grub-0.97-os.1.diff
root@rtl14:/home/L4/grub-0.97/# ./configure;make;make install
```

Note: Before you install the new grub you should check the grub-install version that will be used - if it uses the default grub or only display GRUB 0.97 (that is without the os.1 extension) then something went wrong or you are using the wrong grub version.

So if the result of grub-install -v is:

```
root@rtl14:/home/L4/grub-0.97/#  /usr/local/sbin/grub-install -v
grub-install (GNU GRUB 0.97-os.1)
```

Congratulations! GRUB patching and installation is ok - now then install the modified grub

```
root@rtl14:/home/L4/grub-0.97/# /usr/local/sbin/grub-install /dev/hda
```

Note: We are assume your boot partition is in /dev/hda - also note that GRUB by default searches for a floppy - if you box has no floppy drive this could take quite some time to comlete - `grub-install --help` will list options that could help in that case.

open `/boot/grub/menu.lst` and add the following statements.

3.6.2. Boot L4Linux in L4dope

```
title           L4Linux26/Fiasco+dope
#Assume your boot is in the hda2. It is also (hd0,1)
root (hd0,1)
kernel          /boot/L4Linux/bootstrap
modaddr         0x06000000
module          /boot/L4Linux/main -nowait -nokdb -serial_esc
                                        -comspeed 115200 -comport 1
module          /boot/L4Linux/sigma0
module          /boot/L4Linux/roottask task modname "bmodfs" attached 4 modules
module          /boot/L4Linux/events
```

```
module          /boot/L4Linux/names --events
module          /boot/L4Linux/log --events
module          /boot/L4Linux/dm_phys --isa=0x00800000 -v --events
module          /boot/L4Linux/simple_ts -t 300 --events
module          /boot/L4Linux/rtc --events
module          /boot/L4Linux/l4io --noirq --events
module          /boot/L4Linux/l4exec --events
module          /boot/L4Linux/l4dope --l4io --menubar --transparency
module          /boot/L4Linux/loader --fprov=BMODFS linux26.cfg
module          /boot/L4Linux/bmodfs
module          /boot/L4Linux/vmlinuz26
module          /boot/L4Linux/linux26.cfg
module          /boot/L4Linux/libloader.s.so
module          /boot/L4Linux/libld-l4.s.so
vbeset 0x117 506070
```

(hd0,0) stands for hda1 (firtst ide disk first partition). Similarly, (hd0,1) thus means hda2, (hd1,1) means hdb2 - the numbering is a bit confusing and easy to mess up - but one gets used to it.... save the exit

3.6.3. Boot L4Linux by remote serial console

```
title           L4Linux26/Fiasco+dope
#Assume your boot is in the hda2. It is also (hd0,1)
root            (hd0,1)
kernel          /boot/L4Linux/bootstrap
modaddr         0x06000000
module          /boot/L4Linux/main -nowait -nokdb -serial_esc
                                        -comspeed 115200 -comport 1
module          /boot/L4Linux/sigma0
module          /boot/L4Linux/roottask task modname "bmodfs" attached 4 modules
module          /boot/L4Linux/events
module          /boot/L4Linux/names --events
module          /boot/L4Linux/log --events
module          /boot/L4Linux/dm_phys --isa=0x00800000 -v --events
module          /boot/L4Linux/simple_ts -t 300 --events
module          /boot/L4Linux/rtc --events
module          /boot/L4Linux/l4io --noirq --events module
module /boot/L4Linux/l4exec --events
```

```
module          /boot/L4Linux/loader --fprov=BMODFS linux26.cfg
module          /boot/L4Linux/bmodfs
module          /boot/L4Linux/vmlinuz26
module          /boot/L4Linux/linux26.cfg
module          /boot/L4Linux/libloader.s.so
module          /boot/L4Linux/libld-l4.s.so
```

(hd0,0) stands for hda1 (firtst ide disk first partition). Similarly, (hd0,1) thus means hda2, (hd1,1) means hdb2 - the num bering is a bit confusing and easy to mess up - but one gets used to it.... save the exit

3.7. Collect the components

We create a directory into wich we copy all the binaries, this is strictly speaking not necessary - you could simply set the path to the compile time tree, but we recommend copying it all.

```
root@rtl14:/home/L4/grub-0.97/ mkdir /boot/L4Linux
root@rtl14:/home/L4/grub-0.97/ cd /home/L4/build/bin/x86_586/l4v2/
root@rtl14:/home/L4/build/bin/x86_586/l4v2/
cp {bmodfs,con,dm_phys,events,l4exec,l4io,loader,log,names,roottask,
 rtc,sigma0,simple_ts,libld-l4.s.so,tftp,libloader.s.so,l4dope} /boot/L4Linux/
root@rtl14:/home/L4/build/bin/x86_586/ cp bootstrap /boot/L4Linux/
root@rtl14:/home/L4/l4/ cp kernel/fiasco/build/main /boot/L4Linux/
root@rtl14:/home/L4/grub-0.97/# cp drops_home/l4linux-2.6/vmlinuz26 /boot/L4Linux/
```

3.8. Kernel parameters

After copying it all we need to create the linux configuration, this is a simply script that will be loaded and used to actually launch L4Linux, it is more or less the kernel command line of the Linux kernel.

3.8.1. Boot L4Linux in L4dope

```
root@rtl14:/home/L4/grub-0.97/# vi /boot/L4Linux/linux26.cfg
verbose 0
task "vmlinuz26" "earlyprintk=yes mem=256M video=l4fb root=/dev/hda2"
        all_sects_writable
        allow_cli
```

3.8.2. Boot L4Linux by remote serial console

```
root@rtl14:/home/L4/grub-0.97/# vi /boot/L4Linux/linux26.cfg
modpath "(hd0,1)/boot/L4Linux"
task "vmlinuz26.serialconsole" "earlyprintk=yes console=ttyS1,115200
init=1 mem=256M root=/dev/hda2" all_sects_writable
```

Of course you should adjust this to your settings, the console speed is set to 115200 in our example, not all serial ports like that speed, if you get strange output, check if the other side has the same settings if it does try a lower speed (i.e. 19200). The parameter `mem=256M` tell Fiasco to reserve 256MB for L4Linux, this is a hard upper limit, you might need to adjust this to your system (it makes littel sense to limit L4Linux to 64MB if you box has 512MB of RAM...). The parameter `video=14fb` tells the the L4Linux could use the l4fb not anything else. The last parameter is the well known `root=` which simply tels linux where its root partition is.

```
root@rtl14:/home/L4/grub-0.97/# cat /etc/fstab
         /dev/hda1       /              ext3    defaults,errors=remount-ro 0       1
         /dev/hda5       none           swap    sw                 0       0
```

3.9. Use L4/Fiasco/L4Linux

3.9.1. Use the L4Linux in L4dope

It is very simple. When the l4dope starts you could see that the L4Linux boot in a window. Then you could login in the L4Linux and work with command interface.

3.9.2. Use the L4Linux by remote serial console

When you boot the system, you must modify the run level. If your system is in the X-window when you reboot and choose the L4Linux with dope the L4Linux will try boot the X-windows. It is very slow so I suggest you modify the run level at first.As for Slackware you change the runlevel to 3 and As for Debian or Ubuntu you change it to 1. We will use the serial port 1 (ttyS0) to receive the output from L4/Fiasco - this off course needs to fit your configuration. In common we could check out the system runtime situation through the ttyS0 output. It is like a debug console.

All thats left to do is to reboot the system and choose the L4Linux26/Fiasco+dope at the grub menu.

3.10. L4/Fiasco/L4Linux/L4dope booting log

This part is new and former I didn't take account for the log problem. But when I study it deeper and deeper I find that the log will be more and more important. As for a developer when you encouter a problem the first idea is to check out the log to find out what heppens. If you have a complete log you could identify the problem belongs to the L4Linux or any other modules of drops because if you only see the graphical user interface you find nothing helpful. This log is complete for booting the l4linux and l4dope. If you encouter a problem you could compare these two logs to check out what your problem is. As a newbie encoutering the problem is not a problem. If meeting a problem you can't know what to do, it is a real problem.

```
Enabling special fully nested mode for PIC
Using the PIT (i8254) on IRQ 0 for scheduling
SERIAL ESC: allocated IRQ 4 for serial uart
SERIAL ESC: allocated IRQ 4 for serial uart
Not using serial hack in slow timer handler.
Absolute KIP Syscalls using: Sysenter
CPU: AuthenticAMD (6:8:1:0) Model: Duron (Applebred) at 1601 MHz

  16/256 Entry I TLB (4K pages)      8 Entry I TLB (4M pages)
  32/256 Entry D TLB (4K pages)      8 Entry D TLB (4M pages)
  64 KB L1 I Cache (2-way associative, 64 bytes per line)
  64 KB L1 D Cache (2-way associative, 64 bytes per line)
  64 KB L2 U Cache (8-way associative, 64 bytes per line)

Freeing init code/data: 20480 bytes (5 pages)

SIGMA0: Hello!
  Found Fiasco: KIP syscalls: yes.
  Allocated 188kB for maintenance structures.


Roottask.
  Command line found: "(hd0,1)/boot/L4Linux/roottask task modname

                      "bmodfs" attached 4 modules"

  491071kB ( 479MB) total RAM (reported by bootloader)
  426080kB ( 417MB) received RAM from Sigma0
   25104kB (  25MB) reserved RAM for RMGR
```

```
  Received I/O ports 0000-ffff
  Attached irqs = [ <!0> 1 <!2> 3 <!4> 5 6 7 8 9 A B C D E F ]

Roottask: Parsing command line config.
  configured task 0x00 (bmodfs):
    vm\_offs:0 irq:ffff lmcp:ffff allow\_cli:0 mcp:ff prio:10 small:ff mods:4

Roottask: Loading 15 modules.
#05: loading "(hd0,1)/boot/L4Linux/events"
     from [06113000-06155b12] to [01540000-01549204][0154a000-0154c8f4]
     entry at 00058374 via trampoline page code
     symbols at [1b959000-1b95b000] (8kB), lines at [1b952000-1b959000] (28kB)
#06: loading "(hd0,1)/boot/L4Linux/names --events"
     from [06156000-06195f80] to [002d0000-002d614a][002d7000-002e3000]
     entry at 0005937c via trampoline page code
     symbols at [1b950000-1b952000] (8kB), lines at [1b94a000-1b950000] (24kB)
#07: loading "(hd0,1)/boot/L4Linux/log --events"
     from [06196000-061d1855] to [00400000-00406c6a][00407000-00439850]
     entry at 0005a37c via trampoline page code
     symbols at [1b948000-1b94a000] (8kB), lines at [1b942000-1b948000] (24kB)
#08: loading "(hd0,1)/boot/L4Linux/dm\_phys --isa=0x00800000 -v --events"
     from [061d2000-0625833e] to [01500000-01511d6a][01512000-0151c000]
     entry at 0005b394 via trampoline page code
     symbols at [1b93d000-1b942000] (20kB), lines at [1b930000-1b93d000] (52kB)
#09: loading "(hd0,1)/boot/L4Linux/simple\_ts -t 300 --events"
     from [06259000-062a23e0] to [01400000-0140852a][01409000-0141b030]
     entry at 0005c388 via trampoline page code
     symbols at [1b92e000-1b930000] (8kB), lines at [1b927000-1b92e000] (28kB)
#0a: loading "(hd0,1)/boot/L4Linux/rtc --events"
     from [062a3000-062d3178] to [01080000-01084200][01085000-0108a000]
     entry at 0005e37c via trampoline page code
     symbols at [1b925000-1b927000] (8kB), lines at [1b920000-1b925000] (20kB)
#0b: loading "(hd0,1)/boot/L4Linux/l4io --noirq --events"
     from [062d4000-064b6092] to [00b80000-00ba255c][00ba3000-00bef000]
     entry at 0005f384 via trampoline page code
     symbols at [1b8e8000-1b920000] (224kB), lines at [1b8ce000-1b8e8000] (104kB)
#0c: loading "(hd0,1)/boot/L4Linux/l4exec --events"
     from [064b7000-065d9fe7] to [01100000-01128984][01129000-0114b000]
     entry at 00060380 via trampoline page code
     symbols at [1b8c6000-1b8ce000] (32kB), lines at [1b8a9000-1b8c6000] (116kB)
#0d: loading "(hd0,1)/boot/L4Linux/l4dope --l4io --menubar --transparency"
```

```
     from [065da000-06746350] to [01b00000-01b41010][01b42000-01b8c000]
     entry at 00061394 via trampoline page code
     symbols at [1b89b000-1b8a9000] (56kB), lines at [1b86e000-1b89b000] (180kB)
#0e: loading "(hd0,1)/boot/L4Linux/loader --fprov=BMODFS linux26.cfg"
     from [06747000-06819fbf] to [01300000-01323b30][01324000-0134d000]
     entry at 00062390 via trampoline page code
     symbols at [1b868000-1b86e000] (24kB), lines at [1b84f000-1b868000] (100kB)
#0f: loading "(hd0,1)/boot/L4Linux/bmodfs"
     from [0681a000-068b7615] to [01220000-0123508c][01236000-01256000]
     passing module (hd0,1)/boot/L4Linux/vmlinuz26        [ 068b8000-076e186c ]
     passing module (hd0,1)/boot/L4Linux/linux26.cfg      [ 076e2000-076e2104 ]
     passing module (hd0,1)/boot/L4Linux/libloader.s.so   [ 076e3000-07781afc ]
     passing module (hd0,1)/boot/L4Linux/libld-l4.s.so    [ 07782000-077f83b2 ]
     entry at 00063440 via trampoline page code
     symbols at [1b84a000-1b84f000] (20kB), lines at [1b839000-1b84a000] (68kB)

log     | (hd0,1)/boot/L4Linux/log: unrecognized option `--events'
log     | Unrecognized option: - ?
log     | Usage: (hd0,1)/boot/L4Linux/log <options>. Option list:
log     |  [ -h | --help ]          - this help
log     |  [ -v | --verbose ]       - verbose mode
log     |  [ -l | --local ]         - flush to local console
log     |  [ -L | --nolocal ]       - do not flush to local console
log     |  [ -s | --comport num ]   - flush to specified serial interface (0)
log     |  [ -e | --serial-esc ]    - enter kdebug on esc on serial
log     |  [ -b | --buffer num ]    - buffered mode (0)
log     |  [ --flushprio num ]      - priority of flusher thread (0x20)
log     |  [ -p | --prio num ]      - priority of main thread (0x20)
names   | Starting thread listening for `exit' events
simplets| Configured for 300 tasks.
rtc     | Date:16.11.2006 Time:15:05:15
DMphys  | DMphys memory map:
DMphys  |   phys. memory 0x00000000-0x1dff0000 (from L4 kernel info page)
DMphys  |   using 0x00100000-0x1dff0000
DMphys  |        Memory area      Pool  PS  Flags
DMphys  |   0x00000000-0x00100000  --   --  RESERVED
DMphys  |   0x00100000-0x00134000   7   12  MAPPED
DMphys  |   0x00134000-0x001cb000  --   --  DENIED
DMphys  |   0x001cb000-0x002d0000   7   12  MAPPED
DMphys  |   0x002d0000-0x002e3000  --   --  DENIED
DMphys  |   0x002e3000-0x00400000   7   12  MAPPED
```

```
DMphys  |   0x00400000-0x0043a000  --   --  DENIED
DMphys  |   0x0043a000-0x009e4000   7   12  MAPPED
DMphys  |   0x009e4000-0x00b80000   0   12  MAPPED
DMphys  |   0x00b80000-0x00bef000  --   --  DENIED
DMphys  |   0x00bef000-0x00c00000   0   12  MAPPED
DMphys  |   0x00c00000-0x01000000   0   22  MAPPED
DMphys  |   0x01000000-0x01080000   0   12  MAPPED
DMphys  |   0x01080000-0x0108a000  --   --  DENIED
DMphys  |   0x0108a000-0x01100000   0   12  MAPPED
DMphys  |   0x01100000-0x0114b000  --   --  DENIED
DMphys  |   0x0114b000-0x01220000   0   12  MAPPED
DMphys  |   0x01220000-0x01256000  --   --  DENIED
DMphys  |   0x01256000-0x01300000   0   12  MAPPED
DMphys  |   0x01300000-0x0134d000  --   --  DENIED
DMphys  |   0x0134d000-0x01400000   0   12  MAPPED
DMphys  |   0x01400000-0x0141c000  --   --  DENIED
DMphys  |   0x0141c000-0x01500000   0   12  MAPPED
DMphys  |   0x01500000-0x0151b000  --   --  RESERVED
DMphys  |   0x0151b000-0x01540000   0   12  MAPPED
DMphys  |   0x01540000-0x0154d000  --   --  DENIED
DMphys  |   0x0154d000-0x01b00000   0   12  MAPPED
DMphys  |   0x01b00000-0x01b8c000  --   --  DENIED
DMphys  |   0x01b8c000-0x01c00000   0   12  MAPPED
DMphys  |   0x01c00000-0x06800000   0   22  MAPPED
DMphys  |   0x06800000-0x068b8000   0   12  MAPPED
DMphys  |   0x068b8000-0x077f9000  --   --  DENIED
DMphys  |   0x077f9000-0x07800000   0   12  MAPPED
DMphys  |   0x07800000-0x1b800000   0   22  MAPPED
DMphys  |   0x1b800000-0x1b839000   0   12  MAPPED
DMphys  |   0x1b839000-0x1dff0000  --   --  DENIED
DMphys  |
DMphys  | DMphys memory pools:
DMphys  |   pool 0 (Default memory pool):
DMphys  |   size: 422900KB total, 422900KB free,  32KB reserved
DMphys  |     0x009e4000-0x00b80000 (  1648KB,    2MB)
DMphys  |     0x00bef000-0x01080000 (  4676KB,    5MB)
DMphys  |     0x0108a000-0x01100000 (   472KB,    0MB)
DMphys  |     0x0114b000-0x01220000 (   852KB,    1MB)
DMphys  |     0x01256000-0x01300000 (   680KB,    1MB)
DMphys  |     0x0134d000-0x01400000 (   716KB,    1MB)
DMphys  |     0x0141c000-0x01500000 (   912KB,    1MB)
```

```
DMphys  |     0x0151b000-0x01540000 (   148KB,    0MB)
DMphys  |     0x0154d000-0x01b00000 (  5836KB,    6MB)
DMphys  |     0x01b8c000-0x068b8000 ( 79024KB,   77MB)
DMphys  |     0x077f9000-0x1b839000 (327936KB,  320MB)
DMphys  |   pool 7 (ISA DMA memory pool):
DMphys  |   size:   8192KB total,   8192KB free,   0KB reserved
DMphys  |     0x00100000-0x00134000 (   208KB,    0MB)
DMphys  |     0x001cb000-0x002d0000 (  1044KB,    1MB)
DMphys  |     0x002e3000-0x00400000 (  1140KB,    1MB)
DMphys  |     0x0043a000-0x009e4000 (  5800KB,    6MB)
l4dope  | DOpE(init): using L4 IO server
l4dope  | (hd0,1)/boot/L4Linux/l4dope: unrecognized option `--menubar'
l4dope  | DOpE(init): unknown option!
l4dope  | DOpE(init): using transparency
io      | do\_args(): Disabling internal IRQ handling.
io      | do\_args(): Enabling events support.
io      | CPU supports 8 MTRRs. Allocated:
io      |   0: 00000000-20000000 ( 512MB) type WB
io      |   1: 1e000000-20000000 (  32MB) type UC
io      |   2: e8000000-ec000000 (  64MB) type WC
io      |   3: e8000000-ec000000 (  64MB) type WC
io      | PCI: Using configuration type 1
io      | PCI: Probing PCI hardware
io      | PCI: Probing PCI hardware (bus 00)
io      | PCI: Via IRQ fixup
bmodfs  | Passed the following modules:
bmodfs  |   module "(hd0,1)/boot/L4Linux/vmlinuz26" (14503kB)
bmodfs  |   module "(hd0,1)/boot/L4Linux/linux26.cfg" (1kB)
bmodfs  |   module "(hd0,1)/boot/L4Linux/libloader.s.so" (635kB)
bmodfs  |   module "(hd0,1)/boot/L4Linux/libld-l4.s.so" (473kB)
io      | 00000000-ffffffff : PCI mem
io      |   e0000000-e7ffffff : PCI Bus #01
io      |     e0000000-e7ffffff : S3 Inc. VT8375 [ProSavage8 KM266/KL266]
io      |   e8000000-ebffffff : VIA Technologies, Inc. VT8375 [KM266/KL266] Hos
io      : t Bridge
io      |   ec000000-ec0fffff : PCI Bus #01
io      |     ec000000-ec07ffff : S3 Inc. VT8375 [ProSavage8 KM266/KL266]
io      |   ec100000-ec1000ff : Realtek Semiconductor Co., Ltd. RTL-8139/8139C/
io      : 8139C+
io      |   ec101000-ec1017ff : VIA Technologies, Inc. IEEE 1394 Host Controlle
io      : r
```

```
io      |   ec102000-ec1020ff : VIA Technologies, Inc. USB 2.0
io      | 0000-ffff : PCI IO
io      |   0cf8-0cff : PCI conf1
io      |   d000-d0ff : Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+
io      |   d400-d47f : VIA Technologies, Inc. IEEE 1394 Host Controller
io      |   d800-d81f : VIA Technologies, Inc. USB
io      |   dc00-dc1f : VIA Technologies, Inc. USB (#2)
io      |   e000-e01f : VIA Technologies, Inc. USB (#3)
io      |   e400-e40f : VIA Technologies, Inc. VT82C586A/B/VT82C686/A/B/VT823x/
io      : A/C PIPC Bus Master IDE
io      |   e800-e8ff : VIA Technologies, Inc. VT8233/A/8235 AC97 Audio Control
io      : ler
loader  | vmlinuz26: Starting application using libld-l4.s.so
loader  | vmlinuz26,#10: Loading binary
l4dope  | L4INPUT:                !!! W A R N I N G !!!
l4dope  | L4INPUT:  Please, do not use Fiasco's "-esc" with L4INPUT.
l4dope  | L4INPUT:                !!! W A R N I N G !!!
loader  | vmlinuz26,#10: Loading ldso
loader  | vmlinuz26,#10: Starting libld-l4.s.so at 00013870 via 0000bc54
l4dope  | serio: i8042 AUX port at 0x60,0x64 irq 12
l4dope  | serio: i8042 KBD port at 0x60,0x64 irq 1
l4dope  | input: ImPS/2 Generic Wheel Mouse on isa0060/serio1
l4lx    | ======> L4Linux 2.6 starting... <========
l4lx    | Linux version 2.6.17-l4 (root@fiasco-desktop) (gcc version 3.4.6 (Ubu
l4lx    : ntu 3.4.6-1ubuntu2)) #5 Thu Nov 16 22:52:26 CST 2006
l4lx    | Binary name: vmlinuz26
l4lx    | Kernel command line (4 args): earlyprintk=yes mem=256M video=l4fb roo
l4lx    : t=/dev/hda2
l4lx    | Image: 00400000 - 00731000 [3268 KiB].
l4lx    | Areas: Text:     00400000 - 00642000 [2312kB] (a bit longer)
l4lx    |        Data:     00642000 - 0068bd8c [295kB]
l4lx    |        Initdata: 0068e000 - 006bf000 [196kB]
l4lx    |        BSS:      006c0000 - 0072f6f0 [445kB]
l4lx    | l4lx\_thread\_create: Created thread 10.03 (tamer)
l4lx    | Using tamed mode.
l4lx    | l4env\_linux\_startup thread 4.
l4lx    | l4lx\_thread\_create: Created thread 10.04 (server)
l4lx    | main thread will be 10.04
l4lx    | l4env\_register\_pointer\_section: addr = 00642000 size = 512000
l4lx    |      sec-w-init: virt: 0x00642000 to 0x006befff [500 KiB]
l4lx    |      sec-w-init: Number of physical regions: 1, 512000 Bytes
```

```
l4lx    |      sec-w-init: 1: Phys: 0x0142b000 to 0x014a8000, Size:   512000
l4lx    | main thread: received startup message.
l4lx    | Main thread running, waiting...
l4lx    | setup\_l4env\_memory: Forcing superpages for main memory
l4lx    | Main memory size: 256MB
l4lx    | Got 2048kB of ISA DMA memory.
l4lx    |  ISA DMA memory: virt: 0x00800000 to 0x009fffff [2048 KiB]
l4lx    |  ISA DMA memory: Number of physical regions: 1, 2097152 Bytes
l4lx    |  ISA DMA memory: 1: Phys: 0x0043a000 to 0x0063a000, Size:  2097152
l4lx    |     Main memory: virt: 0x00c00000 to 0x10bfffff [262144 KiB]
l4lx    |     Main memory: Number of physical regions: 1, 268435456 Bytes
l4lx    |     Main memory: 1: Phys: 0x07800000 to 0x17800000, Size: 268435456
l4lx    | Filling lower ptabs...
l4lx    | Done (1611 entries).
l4lx    | l4lx\_thread\_create: Created thread 10.05 (timer.i0)
l4lx    | l4lx\_thread\_create: Created thread 10.06 (Idler)
l4dope  | input: AT Translated Set 2 keyboard on isa0060/serio0
l4dope  | input: PC Speaker
l4dope  | input: L4 input event injector
l4dope  | L4 input event injector: IRQ handler up
l4dope  | connect "ImPS/2 Generic Wheel Mouse", isa0060/serio1/input0
l4dope  | connect "AT Translated Set 2 keyboard", isa0060/serio0/input0
l4dope  | connect "PC Speaker", isa0061/input0
l4dope  | connect "L4 input event injector", l4/sys
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=60, size=800
l4dope  | SharedMem(get\_adr): address = 6000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=61, size=800
l4dope  | SharedMem(get\_adr): address = 7000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=62, size=800
l4dope  | SharedMem(get\_adr): address = d000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=63, size=800
l4dope  | SharedMem(get\_adr): address = e000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=64, size=800
l4dope  | SharedMem(get\_adr): address = f000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=66, size=800
l4dope  | SharedMem(get\_adr): address = 16000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=67, size=800
l4dope  | SharedMem(get\_adr): address = 17000
l4dope  | dope: addr=e0000000 size=31680KiB
io      | Remapping I/O memory e0000000-e1ef0000 cached
io      | Setting MTRR 4 to e0000000-e2000000 type WC
```

```
l4dope  | Mapped video memory at e0000000 to 04000000+000000 [31680kB] via L4IO
l4dope  | mapping: vaddr=0x4000000 size=32440320(0x1ef0000) offset=0(0x0)
l4dope  | Frame buffer base:  0x4000000
l4dope  | Resolution:         1024x768x16
l4dope  | Bytes per scanline: 2048
l4dope  | Current video mode is 1024x768 red=11:5 green=5:6 blue=0:5 res=0:0
l4lx    | Starting L4FB via DOpE
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=75, size=96000
l4dope  | SharedMem(get\_adr): address = 400000
l4lx    | l4lx\_thread\_create: Created thread 10.08 (DOpE refresher)
l4lx    | l4lx\_thread\_create: Created thread 10.09 (L4DOpEinput)
l4lx    | l4lx\_thread\_create: Created thread 10.0a (IRQ17)
l4lx    | l4lx\_thread\_create: Created thread 10.0b (IRQ16)
l4lx    | l4lx\_thread\_create: Created thread 10.0c (IRQ15)
l4lx    | l4lx\_thread\_create: Created thread 10.0d (IRQ14)
l4lx    | l4lx\_thread\_create: Created thread 10.0e (IRQ13)
l4lx    | l4lx\_thread\_create: Created thread 10.0f (IRQ11)
l4lx    | l4lx\_thread\_create: Created thread 10.10 (IRQ10)
l4lx    | l4lx\_thread\_create: Created thread 10.11 (IRQ9)
l4lx    | l4lx\_thread\_create: Created thread 10.12 (IRQ8)
l4lx    | l4lx\_thread\_create: Created thread 10.13 (IRQ7)
l4lx    | l4lx\_thread\_create: Created thread 10.14 (IRQ6)
l4lx    | l4lx\_thread\_create: Created thread 10.15 (IRQ5)
l4lx    | l4lx\_thread\_create: Created thread 10.16 (IRQ3)

Welcome to Fiasco(ia32)!
DD-L4(v2)/x86 microkernel (C) 1998-2006 TU Dresden
Rev: Mon Aug 28 08:34:51 2006 compiled with gcc 3.4.6 for AMD Athlon
Performance-critical config option(s) detected:
  CONFIG\_ASSEMBLER\_IPC\_SHORTCUT is off
  CONFIG\_NDEBUG is off

Enabling special fully nested mode for PIC
Using the PIT (i8254) on IRQ 0 for scheduling
SERIAL ESC: allocated IRQ 4 for serial uart
SERIAL ESC: allocated IRQ 4 for serial uart
Not using serial hack in slow timer handler.
Absolute KIP Syscalls using: Sysenter
CPU: AuthenticAMD (6:8:1:0) Model: Duron (Applebred) at 1601 MHz

  16/256 Entry I TLB (4K pages)      8 Entry I TLB (4M pages)
```

```
  32/256 Entry D TLB (4K pages)      8 Entry D TLB (4M pages)
  64 KB L1 I Cache (2-way associative, 64 bytes per line)
  64 KB L1 D Cache (2-way associative, 64 bytes per line)
  64 KB L2 U Cache (8-way associative, 64 bytes per line)

Freeing init code/data: 20480 bytes (5 pages)

SIGMA0: Hello!
  Found Fiasco: KIP syscalls: yes.
  Allocated 188kB for maintenance structures.


Roottask.
  Command line found: "(hd0,1)/boot/L4Linux/roottask task modname "bmodfs" attached 4 mod

  491071kB ( 479MB) total RAM (reported by bootloader)
  426152kB ( 417MB) received RAM from Sigma0
   25032kB (  25MB) reserved RAM for RMGR
  Received I/O ports 0000-ffff
  Attached irqs = [ <!0> 1 <!2> 3 <!4> 5 6 7 8 9 A B C D E F ]

Roottask: Parsing command line config.
  configured task 0x00 (bmodfs):
    vm\_offs:0 irq:ffff lmcp:ffff allow\_cli:0 mcp:ff prio:10 small:ff mods:4

Roottask: Loading 15 modules.
#05: loading "(hd0,1)/boot/L4Linux/events"
     from [06113000-06155b12] to [01540000-01549204][0154a000-0154c8f4]
     entry at 00058374 via trampoline page code
     symbols at [1b959000-1b95b000] (8kB), lines at [1b952000-1b959000] (28kB)
#06: loading "(hd0,1)/boot/L4Linux/names --events"
     from [06156000-06195f80] to [002d0000-002d614a][002d7000-002e3000]
     entry at 0005937c via trampoline page code
     symbols at [1b950000-1b952000] (8kB), lines at [1b94a000-1b950000] (24kB)
#07: loading "(hd0,1)/boot/L4Linux/log --events"
     from [06196000-061d1855] to [00400000-00406c6a][00407000-00439850]
     entry at 0005a37c via trampoline page code
     symbols at [1b948000-1b94a000] (8kB), lines at [1b942000-1b948000] (24kB)
#08: loading "(hd0,1)/boot/L4Linux/dm\_phys --isa=0x00800000 -v --events"
     from [061d2000-0625833e] to [01500000-01511d6a][01512000-0151c000]
     entry at 0005b394 via trampoline page code
```

```
     symbols at [1b93d000-1b942000] (20kB), lines at [1b930000-1b93d000] (52kB)
#09: loading "(hd0,1)/boot/L4Linux/simple\_ts -t 300 --events"
     from [06259000-062a23e0] to [01400000-0140852a][01409000-0141b030]
     entry at 0005c388 via trampoline page code
     symbols at [1b92e000-1b930000] (8kB), lines at [1b927000-1b92e000] (28kB)
#0a: loading "(hd0,1)/boot/L4Linux/rtc --events"
     from [062a3000-062d3178] to [01080000-01084200][01085000-0108a000]
     entry at 0005e37c via trampoline page code
     symbols at [1b925000-1b927000] (8kB), lines at [1b920000-1b925000] (20kB)
#0b: loading "(hd0,1)/boot/L4Linux/l4io --noirq --events"
     from [062d4000-064b6092] to [00b80000-00ba255c][00ba3000-00bef000]
     entry at 0005f384 via trampoline page code
     symbols at [1b8e8000-1b920000] (224kB), lines at [1b8ce000-1b8e8000] (104kB)
#0c: loading "(hd0,1)/boot/L4Linux/l4exec --events"
     from [064b7000-065d9fe7] to [01100000-01128984][01129000-0114b000]
     entry at 00060380 via trampoline page code
     symbols at [1b8c6000-1b8ce000] (32kB), lines at [1b8a9000-1b8c6000] (116kB)
#0d: loading "(hd0,1)/boot/L4Linux/l4dope --l4io --menubar --transparency"
     from [065da000-06746350] to [01b00000-01b41010][01b42000-01b8c000]
     entry at 00061394 via trampoline page code
     symbols at [1b89b000-1b8a9000] (56kB), lines at [1b86e000-1b89b000] (180kB)
#0e: loading "(hd0,1)/boot/L4Linux/loader --fprov=BMODFS linux26.cfg"
     from [06747000-06819fbf] to [01300000-01323b30][01324000-0134d000]
     entry at 00062390 via trampoline page code
     symbols at [1b868000-1b86e000] (24kB), lines at [1b84f000-1b868000] (100kB)
#0f: loading "(hd0,1)/boot/L4Linux/bmodfs"
     from [0681a000-068b7615] to [01220000-0123508c][01236000-01256000]
     passing module (hd0,1)/boot/L4Linux/vmlinuz26        [ 068b8000-076cfc17 ]
     passing module (hd0,1)/boot/L4Linux/linux26.cfg      [ 076d0000-076d0104 ]
     passing module (hd0,1)/boot/L4Linux/libloader.s.so   [ 076d1000-0776fafc ]
     passing module (hd0,1)/boot/L4Linux/libld-l4.s.so    [ 07770000-077e63b2 ]
     entry at 00063440 via trampoline page code
     symbols at [1b84a000-1b84f000] (20kB), lines at [1b839000-1b84a000] (68kB)

log     | (hd0,1)/boot/L4Linux/log: unrecognized option ‘--events’
log     | Unrecognized option: - ?
log     | Usage: (hd0,1)/boot/L4Linux/log <options>. Option list:
log     |  [ -h | --help ]          - this help
log     |  [ -v | --verbose ]       - verbose mode
log     |  [ -l | --local ]         - flush to local console
log     |  [ -L | --nolocal ]       - do not flush to local console
```

```
log     |  [ -s | --comport num ]   - flush to specified serial interface (0)
log     |  [ -e | --serial-esc ]    - enter kdebug on esc on serial
log     |  [ -b | --buffer num ]    - buffered mode (0)
log     |  [ --flushprio num ]      - priority of flusher thread (0x20)
log     |  [ -p | --prio num ]      - priority of main thread (0x20)
names   | Starting thread listening for ‘exit’ events
simplets| Configured for 300 tasks.
rtc     | Date:17.11.2006 Time:01:51:33
DMphys  | DMphys memory map:
DMphys  |   phys. memory 0x00000000-0x1dff0000 (from L4 kernel info page)
DMphys  |   using 0x00100000-0x1dff0000
DMphys  |        Memory area      Pool  PS  Flags
DMphys  |   0x00000000-0x00100000  --   --  RESERVED
DMphys  |   0x00100000-0x00134000   7   12  MAPPED
DMphys  |   0x00134000-0x001cb000  --   --  DENIED
DMphys  |   0x001cb000-0x002d0000   7   12  MAPPED
DMphys  |   0x002d0000-0x002e3000  --   --  DENIED
DMphys  |   0x002e3000-0x00400000   7   12  MAPPED
DMphys  |   0x00400000-0x0043a000  --   --  DENIED
DMphys  |   0x0043a000-0x009e4000   7   12  MAPPED
DMphys  |   0x009e4000-0x00b80000   0   12  MAPPED
DMphys  |   0x00b80000-0x00bef000  --   --  DENIED
DMphys  |   0x00bef000-0x00c00000   0   12  MAPPED
DMphys  |   0x00c00000-0x01000000   0   22  MAPPED
DMphys  |   0x01000000-0x01080000   0   12  MAPPED
DMphys  |   0x01080000-0x0108a000  --   --  DENIED
DMphys  |   0x0108a000-0x01100000   0   12  MAPPED
DMphys  |   0x01100000-0x0114b000  --   --  DENIED
DMphys  |   0x0114b000-0x01220000   0   12  MAPPED
DMphys  |   0x01220000-0x01256000  --   --  DENIED
DMphys  |   0x01256000-0x01300000   0   12  MAPPED
DMphys  |   0x01300000-0x0134d000  --   --  DENIED
DMphys  |   0x0134d000-0x01400000   0   12  MAPPED
DMphys  |   0x01400000-0x0141c000  --   --  DENIED
DMphys  |   0x0141c000-0x01500000   0   12  MAPPED
DMphys  |   0x01500000-0x0151b000  --   --  RESERVED
DMphys  |   0x0151b000-0x01540000   0   12  MAPPED
DMphys  |   0x01540000-0x0154d000  --   --  DENIED
DMphys  |   0x0154d000-0x01b00000   0   12  MAPPED
DMphys  |   0x01b00000-0x01b8c000  --   --  DENIED
DMphys  |   0x01b8c000-0x01c00000   0   12  MAPPED
```

```
DMphys  |   0x01c00000-0x06800000   0   22  MAPPED
DMphys  |   0x06800000-0x068b8000   0   12  MAPPED
DMphys  |   0x068b8000-0x077e7000  --   --  DENIED
DMphys  |   0x077e7000-0x07800000   0   12  MAPPED
DMphys  |   0x07800000-0x1b800000   0   22  MAPPED
DMphys  |   0x1b800000-0x1b839000   0   12  MAPPED
DMphys  |   0x1b839000-0x1dff0000  --   --  DENIED
DMphys  |
DMphys  | DMphys memory pools:
DMphys  |   pool 0 (Default memory pool):
DMphys  |   size: 422972KB total, 422972KB free,  32KB reserved
DMphys  |     0x009e4000-0x00b80000 (  1648KB,    2MB)
DMphys  |     0x00bef000-0x01080000 (  4676KB,    5MB)
DMphys  |     0x0108a000-0x01100000 (   472KB,    0MB)
DMphys  |     0x0114b000-0x01220000 (   852KB,    1MB)
DMphys  |     0x01256000-0x01300000 (   680KB,    1MB)
DMphys  |     0x0134d000-0x01400000 (   716KB,    1MB)
DMphys  |     0x0141c000-0x01500000 (   912KB,    1MB)
DMphys  |     0x0151b000-0x01540000 (   148KB,    0MB)
DMphys  |     0x0154d000-0x01b00000 (  5836KB,    6MB)
DMphys  |     0x01b8c000-0x068b8000 ( 79024KB,   77MB)
DMphys  |     0x077e7000-0x1b839000 (328008KB,  320MB)
DMphys  |   pool 7 (ISA DMA memory pool):
DMphys  |   size:   8192KB total,   8192KB free,   0KB reserved
DMphys  |     0x00100000-0x00134000 (   208KB,    0MB)
DMphys  |     0x001cb000-0x002d0000 (  1044KB,    1MB)
DMphys  |     0x002e3000-0x00400000 (  1140KB,    1MB)
DMphys  |     0x0043a000-0x009e4000 (  5800KB,    6MB)
bmodfs  | Passed the following modules:
bmodfs  |   module "(hd0,1)/boot/L4Linux/vmlinuz26" (14432kB)
bmodfs  |   module "(hd0,1)/boot/L4Linux/linux26.cfg" (1kB)
bmodfs  |   module "(hd0,1)/boot/L4Linux/libloader.s.so" (635kB)
bmodfs  |   module "(hd0,1)/boot/L4Linux/libld-l4.s.so" (473kB)
l4dope  | DOpE(init): using L4 IO server
l4dope  | (hd0,1)/boot/L4Linux/l4dope: unrecognized option `--menubar'
l4dope  | DOpE(init): unknown option!
l4dope  | DOpE(init): using transparency
io      | do\_args(): Disabling internal IRQ handling.
io      | do\_args(): Enabling events support.
io      | CPU supports 8 MTRRs. Allocated:
io      |   0: 00000000-20000000 ( 512MB) type WB
```

```
io      |   1: 1e000000-20000000 (  32MB) type UC
io      |   2: e8000000-ec000000 (  64MB) type WC
io      |   3: e8000000-ec000000 (  64MB) type WC
io      | PCI: Using configuration type 1
io      | PCI: Probing PCI hardware
io      | PCI: Probing PCI hardware (bus 00)
io      | PCI: Via IRQ fixup
io      | 00000000-ffffffff : PCI mem
io      |   e0000000-e7ffffff : PCI Bus #01
io      |     e0000000-e7ffffff : S3 Inc. VT8375 [ProSavage8 KM266/KL266]
io      |   e8000000-ebffffff : VIA Technologies, Inc. VT8375 [KM266/KL266] Hos
io      : t Bridge
io      |   ec000000-ec0fffff : PCI Bus #01
io      |     ec000000-ec07ffff : S3 Inc. VT8375 [ProSavage8 KM266/KL266]
io      |   ec100000-ec1000ff : Realtek Semiconductor Co., Ltd. RTL-8139/8139C/
io      : 8139C+
io      |   ec101000-ec1017ff : VIA Technologies, Inc. IEEE 1394 Host Controlle
io      : r
io      |   ec102000-ec1020ff : VIA Technologies, Inc. USB 2.0
io      | 0000-ffff : PCI IO
io      |   0cf8-0cff : PCI conf1
io      |   d000-d0ff : Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+
io      |   d400-d47f : VIA Technologies, Inc. IEEE 1394 Host Controller
io      |   d800-d81f : VIA Technologies, Inc. USB
io      |   dc00-dc1f : VIA Technologies, Inc. USB (#2)
io      |   e000-e01f : VIA Technologies, Inc. USB (#3)
io      |   e400-e40f : VIA Technologies, Inc. VT82C586A/B/VT82C686/A/B/VT823x/
io      : A/C PIPC Bus Master IDE
io      |   e800-e8ff : VIA Technologies, Inc. VT8233/A/8235 AC97 Audio Control
io      : ler
loader  | vmlinuz26: Starting application using libld-l4.s.so
loader  | vmlinuz26,#10: Loading binary
l4dope  | L4INPUT:                !!! W A R N I N G !!!
l4dope  | L4INPUT:  Please, do not use Fiasco's "-esc" with L4INPUT.
l4dope  | L4INPUT:                !!! W A R N I N G !!!
loader  | vmlinuz26,#10: Loading ldso
loader  | vmlinuz26,#10: Starting libld-l4.s.so at 00013870 via 0000bc54
l4dope  | serio: i8042 AUX port at 0x60,0x64 irq 12
l4dope  | serio: i8042 KBD port at 0x60,0x64 irq 1
l4dope  | input: ImPS/2 Generic Wheel Mouse on isa0060/serio1
l4lx    | ======> L4Linux 2.6 starting... <========
```

```
l4lx    | Linux version 2.6.17-l4 (root@fiasco-desktop) (gcc version 3.4.6 (Ubu
l4lx    : ntu 3.4.6-1ubuntu2)) #1 Fri Nov 17 08:34:02 CST 2006
l4lx    | Binary name: vmlinuz26
l4lx    | Kernel command line (4 args): earlyprintk=yes mem=256M video=l4fb roo
l4lx    : t=/dev/hda2
l4lx    | Image: 00400000 - 0072e000 [3256 KiB].
l4lx    | Areas: Text:     00400000 - 0063f000 [2300kB] (a bit longer)
l4lx    |        Data:     0063f000 - 0068884c [294kB]
l4lx    |        Initdata: 0068c000 - 006c0000 [208kB]
l4lx    |        BSS:      006c0000 - 0072c800 [434kB]
l4lx    | l4lx\_thread\_create: Created thread 10.03 (tamer)
l4lx    | Using tamed mode.
l4lx    | l4env\_linux\_startup thread 4.
l4lx    | l4lx\_thread\_create: Created thread 10.04 (server)
l4lx    | main thread will be 10.04
l4lx    | l4env\_register\_pointer\_section: addr = 0068a000 size = 671744
l4lx    |      sec-w-init: virt: 0x0068a000 to 0x0072dfff [656 KiB]
l4lx    |      sec-w-init: Number of physical regions: 1, 671744 Bytes
l4lx    |      sec-w-init: 1: Phys: 0x01452000 to 0x014f6000, Size:   671744
l4lx    | main thread: received startup message.
l4lx    | Main thread running, waiting...
l4lx    | setup\_l4env\_memory: Forcing superpages for main memory
l4lx    | Main memory size: 256MB
l4lx    | Got 2048kB of ISA DMA memory.
l4lx    |  ISA DMA memory: virt: 0x00800000 to 0x009fffff [2048 KiB]
l4lx    |  ISA DMA memory: Number of physical regions: 1, 2097152 Bytes
l4lx    |  ISA DMA memory: 1: Phys: 0x0043a000 to 0x0063a000, Size:  2097152
l4lx    |     Main memory: virt: 0x00c00000 to 0x10bfffff [262144 KiB]
l4lx    |     Main memory: Number of physical regions: 1, 268435456 Bytes
l4lx    |     Main memory: 1: Phys: 0x07800000 to 0x17800000, Size: 268435456
l4lx    | Filling lower ptabs...
l4lx    | Done (1614 entries).
l4lx    | l4lx\_thread\_create: Created thread 10.05 (timer.i0)
l4lx    | l4lx\_thread\_create: Created thread 10.06 (Idler)
l4dope  | input: AT Translated Set 2 keyboard on isa0060/serio0
l4dope  | input: PC Speaker
l4dope  | input: L4 input event injector
l4dope  | L4 input event injector: IRQ handler up
l4dope  | connect "ImPS/2 Generic Wheel Mouse", isa0060/serio1/input0
l4dope  | connect "AT Translated Set 2 keyboard", isa0060/serio0/input0
l4dope  | connect "PC Speaker", isa0061/input0
```

```
l4dope  | connect "L4 input event injector", l4/sys
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=60, size=800
l4dope  | SharedMem(get\_adr): address = 6000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=61, size=800
l4dope  | SharedMem(get\_adr): address = 7000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=62, size=800
l4dope  | SharedMem(get\_adr): address = d000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=63, size=800
l4dope  | SharedMem(get\_adr): address = e000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=64, size=800
l4dope  | SharedMem(get\_adr): address = f000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=66, size=800
l4dope  | SharedMem(get\_adr): address = 16000
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=67, size=800
l4dope  | SharedMem(get\_adr): address = 17000
l4dope  | dope: addr=e0000000 size=31680KiB
io      | Remapping I/O memory e0000000-e1ef0000 cached
io      | Setting MTRR 4 to e0000000-e2000000 type WC
l4dope  | Mapped video memory at e0000000 to 04000000+000000 [31680kB] via L4IO
l4dope  | mapping: vaddr=0x4000000 size=32440320(0x1ef0000) offset=0(0x0)
l4dope  | Frame buffer base:  0x4000000
l4dope  | Resolution:         1024x768x16
l4dope  | Bytes per scanline: 2048
l4dope  | Current video mode is 1024x768 red=11:5 green=5:6 blue=0:5 res=0:0
l4lx    | Starting L4FB via DOpE
l4dope  | SharedMem(alloc): hl.low=100000, lh.high=80000, id=75, size=96000
l4dope  | SharedMem(get\_adr): address = 400000
l4lx    | l4lx\_thread\_create: Created thread 10.08 (DOpE refresher)
l4lx    | l4lx\_thread\_create: Created thread 10.09 (L4DOpEinput)
l4lx    | l4lx\_thread\_create: Created thread 10.0a (IRQ17)
l4lx    | l4lx\_thread\_create: Created thread 10.0b (IRQ16)
l4lx    | l4lx\_thread\_create: Created thread 10.0c (IRQ15)
l4lx    | l4lx\_thread\_create: Created thread 10.0d (IRQ14)
l4lx    | l4lx\_thread\_create: Created thread 10.0e (IRQ13)
l4lx    | l4lx\_thread\_create: Created thread 10.0f (IRQ11)
l4lx    | l4lx\_thread\_create: Created thread 10.10 (IRQ10)
l4lx    | l4lx\_thread\_create: Created thread 10.11 (IRQ9)
l4lx    | l4lx\_thread\_create: Created thread 10.12 (IRQ8)
l4lx    | l4lx\_thread\_create: Created thread 10.13 (IRQ7)
l4lx    | l4lx\_thread\_create: Created thread 10.14 (IRQ6)
l4lx    | l4lx\_thread\_create: Created thread 10.15 (IRQ5)
```

```
l4lx    | l4lx\_thread\_create: Created thread 10.16 (IRQ3)
```

3.11. Acknowledgment

Thanks for Adam Lackorzynski. At first I asked many many foolish questions in l4 maillist. But he is very patient and every time he could give me a reply very quickly. Thanks for She Kairui, Zhang Wei and Pu Yinqiao of DSLab. They help me to test this kickstart and I know testing is very boring.

3.12. Reference

- Carsten Weinhold, Developing with L4 - Overview and Pointers //
- Fiasco/L4 Buidling and using http://os.inf.tu-dresden.de/fiasco/use.html//
- L4Linux26 Building and using http://os.inf.tu-dresden.de/L4/LinuxOnL4/build-2.6.shtml//
- http://ubuntu-ph.org/node/5 //
- http://www.ubuntux.org/ubuntu//

4. List of Acronyms

ABI - Aplication Binary Interface
API - Aplication Programming Interface
APIC - Advanced Programmable Interrupt Controller
ADEOS - Adaptive Domain Environment for Operating Systems
CVS - Concurrent Versions System
DROPS - Dresden Real-Time Operating Systems
DMA - Direct Memory Access (coprocessor)
ELF - Executable and Link Format
FDL - Free Documentation License
GCC - GNU Compiler Collection
GNU - GNU Not UNIX (recursive accronym)
GRUB - GRand Unified Bootloader
GPL - General Public License
KIP - Kernel Info Page
L4V2 - L4 Specification Version 2
IPC - Inter Process Communication
OS - Operating System
RT - Real Time
RTC - Real Time Clock
UML - User Mode Linux