

## L4/Fiasco kernel build

The steps necessary to build a Fiasco kernel are as follows:

### Get Fiasco and needed modules from our read-only SVN

How to get all the needed modules from our SVN is described on the Get DROPS page.

1. Checkout the hello module to get everything (Recommended):
  1. The hello module contains more than what is actually needed, so don't be scared!
  2. If you're only interested in Fiasco itself (i.e. compile without Roottask, Sigma0, etc.), the following suffices:

```
l4/tool/preprocess
kernel/fiasco
```

1. Don't call make in the l4/tool/preprocess directory, it won't work because of the missing l4 checkout and it's NOT necessary to build Fiasco!
2. Check configuration. (Only needed when choosing "1a")
3. Configure the L4 system libraries and support servers:

```
make -C l4 O=/my-user-build-dir oldconfig
```

1. Most likely you do not need to change anything, so oldconfig is ok. If you want to change something you may also take the config target.
2. /my-user-build-dir is any directory you like. That's the location where any generated file will be put in.
3. Compile the L4 system libraries and support servers.
  1. Compilation when you've checked out the whole L4 tree:

```
cd l4
make O=/my-user-build-dir
```

1. This will build all packages that you have checked out and that are necessary to run Fiasco with some programs.
  1. When building Fiasco only:
2. Nothing needs to be done here, proceed with the next point.
3. Compile Fiasco.
4. First, create a build directory using:

```
cd kernel/fiasco
make BUILDDIR=/my-kernel-build-dir
```

1. The path given to make BUILDDIR= can either be absolute or relative to kernel/fiasco.
2. Now change to your build directory to configure the kernel.

```
cd /my-kernel-build-dir
make menuconfig
```

1. With make menuconfig the kernel configurator will be started. You can now configure your

kernel. Use the 'h' key to get help, use the arrow keys to navigate through the menus and space to switch the options, use 'x' to save and exit.

2. At this stage, your kernel should be properly configured, so you can proceed to building it:

```
make
```

1. This should create a binary named `main` and symbolic link `fiasco` pointing to `main`.
2. An even faster way to build a preconfigured `fiasco` kernel is to go to `kernel/fiasco` and call `make fiasco`:

```
cd kernel/fiasco  
make fiasco
```

1. This will build a `fiasco` kernel with the `ia32-short` template in the `kernel/fiasco/build` directory.
2. Note: If you're looking for `Fiasco-UX` (the `Fiasco` port to Linux User space), you probably want to take a look at its own page.

## Booting Fiasco

To boot `Fiasco`, you need:

- The bootstrap binary can be found at `<builddir>/bin/$(ARCH)/bootstrap`.

`$(ARCH)` is something like `x86_586` and `$(L4API)` is something like `l4v2`.

- The `Fiasco` binary at `/my-kernel-build-dir/fiasco`.
- The `Sigma0` pager. It is the first user-level server started by `Fiasco`. `Sigma0` has been built in step 2 in the previous section. Its binary can be found at `<builddir>/bin/$(ARCH)/$(L4API)/sigma0`.
- The resource manager `Roottask`. `Roottask` is the second server started by `Fiasco`.
- `Roottask` also has been built in step 2 in the previous section. The `Roottask` binary can be found at `<builddir>/bin/$(ARCH)/$(L4API)/roottask`.
- At least one L4 application program. For demonstration purposes, we will use the `Hello World` server which also has been built in step 2 in the previous section and can be found at `<builddir>/bin/$(ARCH)/$(L4API)/hello`.
- The `L4Linux` server is another very popular `Fiasco` application program.
- The `GRUB` bootloader. I won't discuss here how to build and install `GRUB`; for information about that please visit the `GRUB` website.

For the examples below, I'll assume you use the `courtesy GRUB boot floppy` we provide (see the `download page` for details).

A typical `menu.lst` file looks as follows:

```
title hello world  
kernel (fd0)/bootstrap  
modaddr 0x02000000  
module (fd0)/fiasco -nokdb  
module (fd0)/sigma0  
module (fd0)/roottask
```

```
module (fd0)/hello
```

When booted, you should see the hello word server repeatedly printing its thread id.

Some notes:

- If you leave out the "-nokdb" option for Fiasco (see step 3 in the previous section), Fiasco would enter a kernel breakpoint at startup and try to connect to a remote GDB connected via a serial line (COM1).
- Other Fiasco command line options are documented in the Fiasco online manual page.

Download L4Linux

L4Linux is freely available under the GNU General Public License and available via SVN.

Older releases (linux22, linux(20)) are not available anymore. In urgent matters you may contact the webmaster to get access.

To use L4Linux on Fiasco, you also need the L4 development environment as well as Fiasco itself. To get the L4 programs required for L4Linux-2.6 please use the l4linux-2.6-env module. Downloading L4Linux-2.6 source via read-only SVN

The read-only SVN repository is available via HTTP at <http://svn.tudos.org/repos/l4linux/trunk>.

```
svn co http://svn.tudos.org/repos/l4linux/trunk l4linux-2.6
```

This should give you a l4linux-2.6 directory.

From:

<https://osfree.su/doku/> - **osFree wiki**

Permanent link:

<https://osfree.su/doku/doku.php?id=en:docs:kernel:fiasco:build&rev=1400705204>

Last update: **2014/05/21 20:46**

