4. BARRELFISH

This chapter will describe how to compile and run an application in a different OS (Barrelfish) running in GEM5.

The first step is to download the Barrelfish you would like to work with available on (*http://hg.barrelfish.org/*). The revision tested was: *970955217e45*.

Barrelfish configuration:

```
cd <path_to_be_tested>
```

Download:

```
hg clone --rev 970955217e45 http://hg.barrelfish.org/
```

The second step is to create a build directory and compile it to the desired architecture.

Create a build directory:

```
mv hg.barrelfish.org barrelfish
mkdir barrelfish/build
cd barrelfish/build
```

Build Barrelfish:

```
../hake/hake.sh -s .. -a arm_gem5
```

Compiling:

```
make arm
make arm_gem5/romfs.cpio
```

Now is time to create your application. For simplicity reasons, we will copy the *hellotest* structure and modify it.

How to create an application to Barrelfish:

```
cd <path_to_be_tested>/barrelfish/usr/tests
cp -R hellotest <app_name>
cd <app_name>
mv hellotest.c <app_name>.c
```

Open the Hakefile file and change all 'hellotest' to '<app_name>', after you must modify the <app_name>.c to create here your application.

Include on barrelfish/hake/menu.lst.<ARCH>:

```
module /<ARCH>/sbin/<app_name>
```

Include on barrelfish/hake/symbolic_targets.mk after line "sbin/hellotest \":

```
sbin/<app_name> \
```

Finally, after all this modifications, you need to re-build the architecture target kernel to be tested.

The second part is to get the GEM5 simulator and modify it to accept running with Barrelfish. To simplicity reasons, we will use the revision: *d45a02bd5391*.

GEM5 configuration:

```
cd <path_to_be_tested>
```

Download:

```
hg clone --rev d45a02bd5391 http://repo.gem5.org/gem5
cd gem5
```

How the GEM5 is always been updated, you need to apply the following patch`s before you create your environment and copy the script that will be used to create the architecture to the properly directory.

Apply the patch's:

```
<path>/barrelfish/tools/arm_gem5/system.cc.patch -> <path>/gem5/src/arch/arm/linux/system.cc
<path>/barrelfish/tools/arm_gem5/gic.cc.patch -> <path>/gem5/src/dev/arm/gic.cc
<path>/barrelfish/tools/arm_gem5/Realview.cc.patch -> <path>/gem5/src/dev/arm/Realview.cc
```

Copy:

```
cp <path>/barrelfish/tools/arm_gem5/gem5script.py <path>/gem5/configs/example/gem5script.py
```

Update gem5script.py:

```
addToPath('../common')
system.boot_loader = '../barrelfish/tools/arm_gem5/boot.arm'
```

Create an ARM environment:

```
scons build/ARM/gem5.fast
```

After setting up your Barrelfish and GEM5 environment, you can test your application with the following commands:

```
cd <path>/gem5
build/ARM/gem5.opt configs/example/gem5script.py --kernel=../barrelfish/build/arm_gem5/sbin/cpu --
ramdisk=../barrelfish/build/arm_gem5/romfs.cpio
```