Openmul HOWTO –

Add a new application to build system

Contents

1 Create a new application directory and files

```
$ cd openmul/applications
$ mkdir myapp
$ cp -f hello/hello.c myapp/myapp.c
$ cp -f hello/hello.h myapp/myapp.h
$ cp -f  hello/Makefile.am myapp/
$ cd myapp
```

2 Edit application's Makefile.am

```
ACLOCAL_AMFLAGS = -I m4 ${ACLOCAL_FLAGS}

pkgconfigdir=$(libdir)/pkgconfig
bin_PROGRAMS = mulmyapp

LIB_GLIB = @LIB_GLIB@
LIB_EVENT = @LIB_EVENT@
GLIB_INCLUDE = @GLIB_INCLUDE@
LIBEVENT_INCLUDE = @LIBEVENT_INCLUDE@

INCLUDES = -I. -I$(top_srcdir)/mul -I$(top_srcdir)/common/ \
        -I$(top_srcdir)/services/loadable/topo_routing/ \
        -I$(top_srcdir)/common-libs/3rd-party/derived-lib/include/ \
        -I$(top_srcdir)/common-libs/3rd-party/quagga-lib/include/ \
        -I$(top_srcdir)/common-libs/mul-lib/include/

DEFS = @DEFS@ -DSYSCONFDIR=\"$(sysconfdir)/\"

mulhello_SOURCES = myapp.c $(top_srcdir)/common/mul_app_main.c

LD_FLAGS= @LDFLAGS@

if !HAVE_EXT_GLIB
LD_FLAGS += -Wl,-rpath=$(LIB_GLIB) -L$(LIB_GLIB)
INCLUDES += -I$(GLIB_INCLUDE) -I$(GLIB_INCLUDE)/../
endif
if !HAVE_EXT_LIBEVENT
LD_FLAGS += -Wl,-rpath=$(LIB_EVENT) -L$(LIB_EVENT)
INCLUDES += -I$(LIBEVENT_INCLUDE)
endif

mulhello_LDADD = \
            $(top_srcdir)/common-libs/mul-lib/.libs/libmulutil.a  \
            $(top_srcdir)/services/loadable/topo_routing/./.libs/libmultr.a -lrt \
            -lcrypto -lssl -lglib-2.0

mulhello_LDFLAGS = $(LD_FLAGS) -lpthread -levent -lcrypt

noinst_HEADERS = myapp.h
~
```

3 Replace all “HELLO_APP_NAME” to your app name in myapp.c

Example:

```
219    mul_register_app_cb(NULL,              /* Application specific arg */
220                    MY_APP_NAME,      /* Application Name */
221                    C_APP_ALL_SW,       /* Send any switch's notification */
222                    C_APP_ALL_EVENTS,    /* Send all event notification per switch */
223                    0,                /* If any specific dpid filtering is requested */
224                    NULL,               /* List of specific dpids for filtering events */
225                    &hello_app_cbs);     /* Event notifier call-backs */
226
227    return;
228 }
```

All other occurrences of the name “HELLO_APP_NAME” in the application also needs to change.

4 Add a macro to define your app name

Edit/ modify the file : <openmul>/common-libs/mul-lib/include/mul_app_interface.h

```
34 /* Registered application names */
 35 #define HELLO_APP_NAME "mul-hello"
 36 #define MY_APP_NAME "mul-myapp"
 37 #define FAB_APP_NAME "mul-fabric"
 38 #define CLI_APP_NAME "mul-cli"
 39 #define L2SW_APP_NAME "mul-l2sw"
 40 #define TR_APP_NAME "mul-tr"
 41 #define MAKDI_APP_NAME "mul-makdi"
 42 #define FEMTO_APP_NAME "mul-femto"
 43 #define PRISM_APP_NAME "prism"
 44 #define CONX_APP_NAME "ConX"
 45 #define DRONE_APP_NAME "Drone"
 46 #define MUL_MAX_SERVICE_NUM 10
```

5 Edit <openmul>/Makefile.am file and add your application

```
SUBDIRS = \
        docs \
        common-libs/mul-lib/ \
        common-libs/util-linux/libuuid \
        common-libs/3rd-party/quagga-lib \
        mul \
        services/loadable/topo_routing \
```

```
        services/loadable/conx \
        application/hello \
        application/myapp \
        application/l2switch \
        application/makdi \
        application/fabric \
        application/cli \
        application/prism/agent/ \
        application/prism/app/ \
        application/nbapi/c-swig\
        application/drone

DIST_SUBDIRS = \
            docs \
            common-libs/mul-lib/ \
            common-libs/3rd-party/quagga-lib \
            common-libs/util-linux/libuuid \
            mul \
            services/loadable/topo_routing/ \
            services/loadable/conx/ \
            application/hello \
            application/l2switch \
            application/myapp \
            application/makdi \
            application/fabric \
            application/cli \
            application/prism/agent/ \
            application/prism/app/ \
            application/nbapi/c-swig\
            application/drone

EXTRA_DIST = aclocal.m4

ACLOCAL_AMFLAGS = -I m4
```

6 Register your application in GNU automake system. Edit openmul/configure.ac

```
174 AC_CONFIG_FILES([Makefile mul/Makefile application/l2switch/Makefile \
175         application/hello/Makefile docs/Makefile \
176         application/myapp/Makefile \
177         application/drone/Makefile \
178         application/makdi/Makefile application/fabric/Makefile \
179         application/cli/Makefile application/prism/agent/Makefile \
180         application/prism/app/Makefile application/nbapi/c-swig/Makefile \
181         services/loadable/topo_routing/Makefile \
182         services/loadable/conx/Makefile \
183         common-libs/3rd-party/quagga-lib/Makefile \
184         common-libs/mul-lib/Makefile \
185         common-libs/util-linux/libuuid/Makefile])
```

7 Reconfigure and rebuild

```
$ cd <openmul>
$ ./autogen.sh
$ CFLAGS=`pkg-config --cflags glib-2.0` ./configure --with-vty=yes
$ make
```