Getting Started with SH4 and QEMU

Introduction

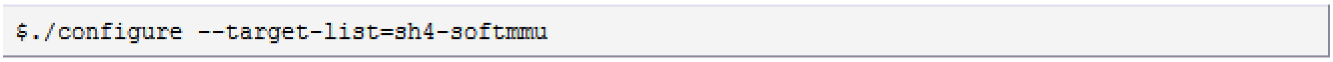
QEMU(http://wiki.qemu.org/Main_Page) is an open source processor emulator supporting many different architectures, including SuperH. This document is derived from QEMU for SH4(http://www.assembla.com/wiki/show/qemu-sh4) originally published in April 2009 by Kawasaki-san.(http://www.assembla.com/profile/kawasaki)

Installing QEMU

- Clone the Master QEMU Git tree:

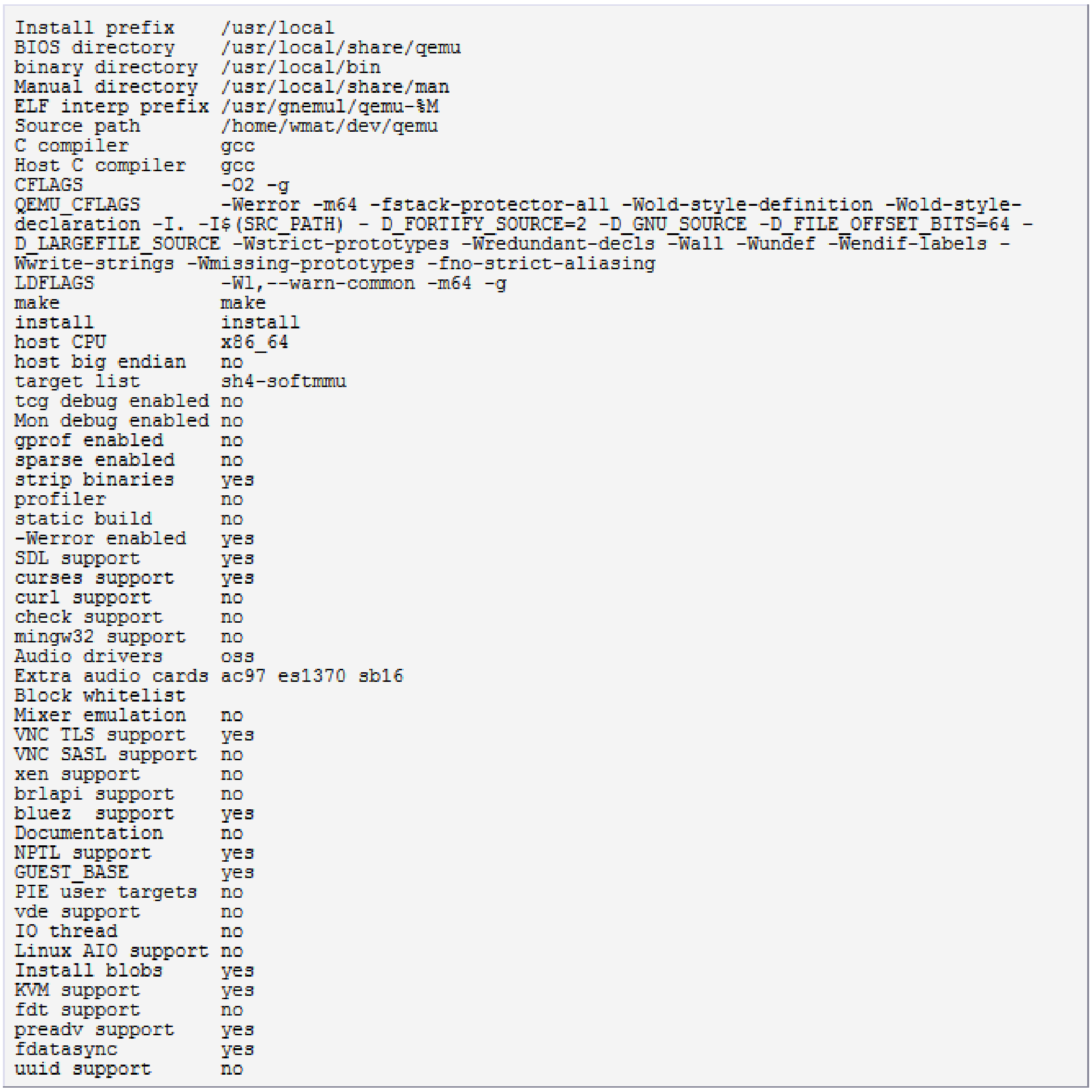
git clone git://git.qemu.org/qemu.git qemu

- Build QEMU for SH4
    - First run the configure script:



```
$./configure --target-list=sh4-softmmu
```

On my Ubuntu9.1 system this results in the following ./configure output:



```
Install prefix    /usr/local
BIOS directory    /usr/local/share/qemu
binary directory  /usr/local/bin
Manual directory  /usr/local/share/man
ELF interp prefix /usr/gnemul/qemu-%M
Source path       /home/wmat/dev/qemu
C compiler        gcc
Host C compiler   gcc
CFLAGS            -O2 -g
QEMU_CFLAGS       -Werror -m64 -fstack-protector-all -Wold-style-definition -Wold-style-
declaration -I. -I$(SRC_PATH) - D_FORTIFY_SOURCE=2 -D_GNU_SOURCE -D_FILE_OFFSET_BITS=64 -
D_LARGEFILE_SOURCE -Wstrict-prototypes -Wredundant-decls -Wall -Wundef -Wendif-labels -
Wwrite-strings -Wmissing-prototypes -fno-strict-aliasing
LDFLAGS           -Wl,--warn-common -m64 -g
make              make
install           install
host CPU          x86_64
host big endian   no
target list       sh4-softmmu
tcg debug enabled no
Mon debug enabled no
gprof enabled     no
sparse enabled    no
strip binaries    yes
profiler          no
static build      no
-Werror enabled   yes
SDL support       yes
curses support    yes
curl support      no
check support     no
mingw32 support   no
Audio drivers     oss
Extra audio cards ac97 es1370 sb16
Block whitelist
Mixer emulation   no
VNC TLS support   yes
VNC SASL support  no
xen support       no
brlapi support    no
bluez  support    yes
Documentation     no
NPTL support      yes
GUEST_BASE        yes
PIE user targets  no
vde support       no
IO thread         no
Linux AIO support no
Install blobs     yes
KVM support       yes
fdt support       no
preadv support    yes
fdatasync         yes
uuid support      no
```

- Next, run make and test:

```
$make
$./sh4-softmmu/qemu-system-sh4
Initializing CPU
Allocating ROM
Allocating SDRAM 1
Allocating SDRAM 2
shix_init: load BIOS 'shix_bios.bin'
ret=-1
qemu: could not load SHIX bios 'shix_bios.bin'
```

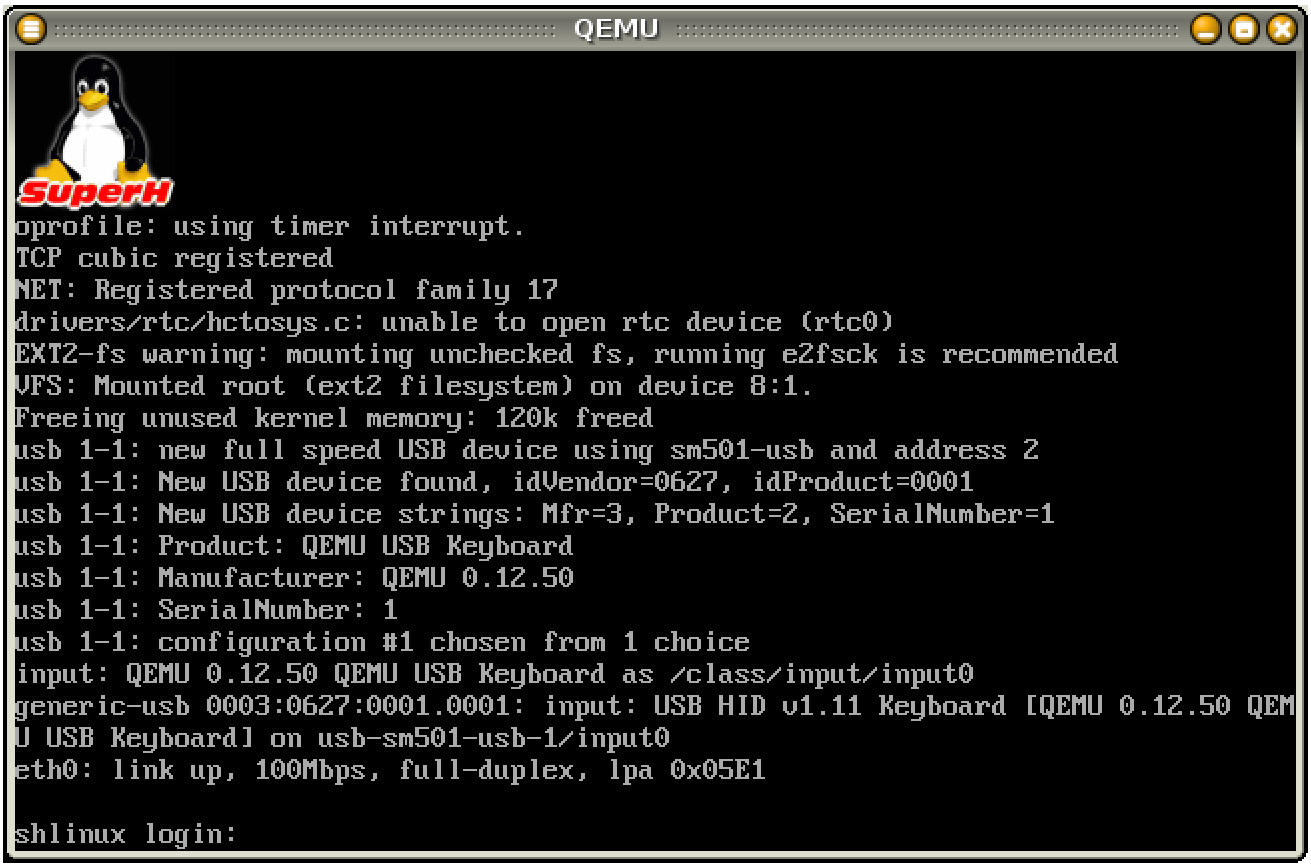
Using Pre-Built Kernel and Userland

It is possible to get quickly started by using a pre-built Linux Kernel and downloadable from here.

Upack sh-test-0.2.tar.bz2 and then invoke QEMU with the following. Note that I am invoking from within my qemu directory. I unpacked the kernel and userspace at the same level as the qemu directory. You may or may not need to adjust your command applicable to your directory structure:

```
./sh4-softmmu/qemu-system-sh4 -M r2d -serial stdio -m 1024M -kernel ../sh-test-0.2/zImage -
usb -usbdevice keyboard -hda ../sh-test-0.2/sh-linux-mini.img
```

The results (on my system) in the following QEMU window:



```
oprofile: using timer interrupt.
TCP cubic registered
NET: Registered protocol family 17
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
EXT2-fs warning: mounting unchecked fs, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem) on device 8:1.
Freeing unused kernel memory: 120k freed
usb 1-1: new full speed USB device using sm501-usb and address 2
usb 1-1: New USB device found, idVendor=0627, idProduct=0001
usb 1-1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
usb 1-1: Product: QEMU USB Keyboard
usb 1-1: Manufacturer: QEMU 0.12.50
usb 1-1: SerialNumber: 1
usb 1-1: configuration #1 chosen from 1 choice
input: QEMU 0.12.50 QEMU USB Keyboard as /class/input/input0
generic-usb 0003:0627:0001.0001: input: USB HID v1.11 Keyboard [QEMU 0.12.50 QEM
U USB Keyboard] on usb-sm501-usb-1/input0
eth0: link up, 100Mbps, full-duplex, lpa 0x05E1

shlinux login:
```