Homework

William Maxwell, Jing Huang, Deval Prashant

May 6, 2018

Abstract

1 Design Plan

First, we create a new file `block/sstf-iosched.c`. We copy the contents of `block/noop-iosched.c` into our new file to use as a template. We do a search and replace to change all instances of *noop* to *sstf* using the command `%s/noop/sstf/g` in `vim`.

Next, we modify the contents of `block/Makefile` so our new scheduler will be compiled. We add the line `obj-$(CONFIG_IOSCHED_SSTF) += sstf-iosched.o`.

We add a configuration option for the new scheduler in the `block/Kconfig.iosched` file. The line `config IOSCHED_SSTF` to create the option, and the line `default "sstf" if DEFAULT_SSTF` is added to make our scheduler a default option.

Now, running `make menuconfig` under the block layer options you will see the new SSTF scheduler. Select it as the default scheduling algorithm and run `make -j4 all`. We also change the `LOCALVERSION` variable to append the string *Group12* to the kernel name. Then we can use `uname -a` to be sure that we've booted into the proper kernel. We can now run QEMU, but we need to make sure it is using the new kernel. We simply change the path for the `-kernel` flag to `linux-yocto-3.19/arch/i386/boot/bzImage`.

```
qemu-system-i386 -gdb tcp::5604 -S -nographic -kernel linux-yocto-3.19/arch/i386/boot/bzImage
-drive file=core-image-lsb-sdk-qemux86.ext4,if=virtio -enable-kvm -net none
-usb -localtime --no-reboot --append "root=/dev/vda rw console=ttyS0 debug"
```

Finally, we create a patch file for the `block` directory with the command `git diff block/ > group12.patch`.

2 Work Log

3 Questions

3.1 What do you think the main point of this assignment is?

There are two purposes to this assignment. The first is to understand the main ideas behind scheduling algorithms by implementing one ourselves. The second is to become familiar with Linux kernel development. This involves reading the documentation for scheduling algorithms and understanding the build process behind compiling your own scheduler.

3.2 How did you personally approach the problem?

We started by reading the Linux elevator source code. Much of this was in `elevator.h`. We identified the functions used to add requests to the request queue and to dispatch requests. These are the functions we needed to modify in order to implement our scheduler.

3.3 How did you ensure your solution was correct?

3.4 What did you learn?

3.5 How should the TA evaluate your work?