



Virginia Tech ❖ Bradley Department of Electrical and Computer Engineering

ECE 4984 / 5984 Linux Kernel Programming
Fall 2017

Large Project: IO Scheduler

1 Introduction

In this project, we will develop a new IO scheduler, named *COOP*, which is a cooperative IO scheduler considering both of request time and process priority. The following concepts from the course will be put in practice in this project:

- Generic block layer
- IO scheduler

2 Project description

COOP should run in kernel space and should be written as a kernel module without modifying Linux kernel source code. All source code should be compiled with Linux Kernel v4.12. Following is the more detail description of the project

Restriction : Do not modify existing kernel source code at all and write COOP as a kernel module.

IO scheduling policy : Design a new IO scheduling policy that considers `vruntime` of a task and the arrival time of an IO request. COOP prefers IO requests, which arrived earlier and were issued by a task with smaller `vruntime`.

Benchmarking : Run `fio` file system benchmark, which emulates access pattern of file servers. The exact configuration for `fio` benchmark will be provided via the course web site.

Analysis : Collect block access traces while running `fio` benchmark with COOP and CFQ using `blktrace`, which is a block trace tool in Linux.

Visualization : Compare block access patterns of COOP and CFQ. Draw two disk trace graphs, whose x-axis is time and y-axis is disk block address. It is okay to use any graph generation tool (e.g., `gnuplot`) other than GUI-based tools, such as Excel or LibreOffice.

3 Additional information

Noop scheduler : <https://github.com/torvalds/linux/blob/master/block/noop-iosched.c>

Deadline scheduler : <https://github.com/torvalds/linux/blob/master/block/deadline-iosched.c>

CFQ scheduler : <https://github.com/torvalds/linux/blob/master/block/cfq-iosched.c>

fio : <https://linux.die.net/man/1/fio>

blktrace : <https://linux.die.net/man/8/blktrace>

gnuplot : <http://www.gnuplotting.org/>

4 Results to be handed

The deadline is **12/06 11:59 PM EDT**. Following is expected to be submitted:

- Source code. It is okay to copy and modify existing kernel code. However, you should comment in the source code to show that you understand it.
- Script to collect block traces and visualize the collected traces.
- Report in PDF, which briefly describes your design and compares block traces of COOP scheduler and CFQ scheduler for fio benchmark.

All of this should be contained in a tarball should, with the following format: `<group id>.project5.tar.gz` (e.g., `group1.project5.tar.gz`)