Project 2 – Round Robin Scheduler

Brent Mitchell & Tyler Yoder

# Introduction

In this project, we implemented a round robin scheduler in the Linux kernel. This was done by modifying the scheduler classes, finishing the implementation of the round robin scheduling class, adding a system call to set the quantum, and analyzing the new kernel build by using a multithreaded program on a custom build on KVM.

# Implementation

The general implementation of the project followed the provided guide. First, sched\_other\_rr.c was modified at several functions, as described below. Then we modified sched\_setscheduler to select our new round robin scheduler, and added a system call for setting the quantum to the three files listed on the guide. Finally, we built and debugged the kernel, and tested it using the threadrunner program.

## Functions

Several sched\_other\_rr.c functions were modified. The modifications are explained briefly below.

### enqueue\_task\_other\_rr

This function adds a task to the end of the run queue. First, the time slice is set to the default quantum. Then the task is added to the run queue. Then the count of running tasks is incremented.

### dequeue\_task\_other\_rr

This function removes a task from the run queue. First the task is removed from the run queue, then the count of running tasks is decremented.

### yield\_task\_other\_rr

This function ends a task’s control of the CPU and adds it to the end of the run queue. This is done by using the requeue\_task\_other\_rr function, which has already been defined in the program.

### pick\_next\_task\_other\_rr

This function selects the next task to run (the head of the run queue). First, it checks if the queue is empty. If it isn’t, the next pointer is set to the run queue’s head item. Then the clock is started on that item.

### task\_tick\_other\_rr

This function decrements the current task’s time slice by one time unit (jiffie). First, it checks if the default quantum is 0. If it is, that means the round robin scheduler is set to behave like FCFS, so the function returns automatically. Otherwise, the time slice is decremented by 1. If the time slice is now 0, then the reschedule flag is set and the current task is yielded.

# Testing

To be completed

# Difficulties in Implementation

To be completed