OS Project 1 Report

B07902085 / 資工二 / 張琪

Design

System Calls

1. `sys_pj1_time`
   return the time from `getnstimeofday` with combined seconds and nanoseconds.
2. `sys_pj1_print`
   receives 3 parameters `(pid, start_time, finish_time)` and printk `KERN_INFO` to dmesg.

process

Struct :

1. `name`
   name of process.
2. `ready_time`
   time when the process is ready.
3. `exec_time`
   time left for the process to finish execution.
   each time the process run, exec_time is decreased by 1.
4. `pid`
   process id.
   id is -1 if a process is not yet ready to run at specific time.

Function :

1. `unit_time`
   one unit time for execution of an empty loop of one million iterations.
2. `proc_assign_cpu`
   assign process to specific CPU by `shced_setaffinity` .
3. `proc_exec`
   execute process and return the process ID.
4. `proc_low_prio`
   set process priority to low by `sched_setscheduler` .
5. `proc_high_prio`
   set process priority to high by `sched_setscheduler` .

scheduler

Variables :

1. `curr_time`
   current unit time.
2. `run_id`
   id of currently running process.
   it is -1 if the there is no running process.
3. `finished`
   total of processes that have finished executing.
4. `last_switch`
   time of last switch for RR scheduling.

Function :

1. `sched_FIFO`
2. `sched_RR`
3. `sched_SJF`
4. `sched_PSJF`

FIFO

In each unit time :

1. Check for processes that are ready to run at that time. Execute ready process and firstly set its priority to low.
2. Look for next process to run.
   - If there is no running process, look for the next ready process in queue.
   - If there is running process, let it run to finish (non-preemptive).
3. For context switch to happen, set priority of current running process to low and set priority of next process to high.

RR

In each unit time :

1. Check for processes that are ready to run at that time. Execute ready process and firstly set its priority to low.
2. Look for next process to run.
   - If there is no running process, look for the next ready process in queue.
   - If there is running process and one quantum time has passed, change the ready time of current running process to current time and put it back to ready queue (preemptive). Then, look for next ready process in queue.
   - If there is running process and one quantum time has not passed, let it run until one quantum time has passed (non-preemptive).
3. For context switch to happen, set priority of current running process to low and set priority of next process to high.

SJF

In each unit time :

1. Check for processes that are ready to run at that time. Execute ready process and firstly set its priority to low.
2. Look for next process to run
    - If there is no running process, look for the next ready process in queue with shortest execution time left.
    - If there is running process, let it run to finish (non-preemptive).
3. For context switch to happen, set priority of current running process to low and set priority of next process to high.

PSJF

In each unit time :

1. Check for processes that are ready to run at that time. Execute ready process and firstly set its priority to low.
2. Look for next process to run
    - If there is no running process, look for the next ready process in queue with shortest execution time left.
    - If there is running process, check if there is any process that is ready with shorter execution time left. If there exists such process, push running process to ready queue and run the other process with shorter execution time (non-preemptive). If there is no such process, let the running process to continue running.
3. For context switch to happen, set priority of current running process to low and set priority of next process to high.

Environment

Kernel version : 4.14.25
Linux version : Ubuntu 16.04

Comparison

- The finish order of processes of this implementation is the same to the theoretical result.
- In implementation, it took a little bit more time in compared to the theoretical result as scheduler has various jobs besides scheduling the processes. For example, it has to set process priority and others data structure related function. As a result, the scheduler speed would be decreased and executing a process will take longer than expected.
- For sure, our computer has to run other processes other than this scheduling program, thus there is a lot more context switch happening in the background which slows down this execution.