Project 4. Priority Scheduler

2019.5.1 (Wed.)

TAs
김종석(ks77sj@gmail.com) / 최동규(gmj03003@gmail.com)
Project Plan

- Total 7 projects
  0) Starting xv6 operating system (5%)
  1) System call (10%)
  2) Thread (15%)
  3) Synchronization (15%)
  4) Scheduling 1 (10%)
  5) Scheduling 2 (15%)
  6) Page fault handler (15%)
  7) Copy on Write (15%)
Xv6 Process

• Process states (procstate in proc.h)
  – UNUSED: Not used
  – EMBRYO: Newly allocated (not ready for running yet)
  – SLEEPING: Waiting for I/O, child process, or time
  – Runnable: Ready to run
  – RUNNING: Running on CPU
  – ZOMBIE: Exited
Xv6 Process Scheduler

- `scheduler()` in `proc.c`
  - Round-robin fashion

```c
void scheduler(void)
{
    struct proc *p;
    for(;;){
        // Enable interrupts on this processor.
        sti();
        // Loop over process table looking for process to run.
        acquire(&ptable.lock);
        for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
            if(p->state != RUNNABLE)
                continue;
            // Switch to chosen process. It is the process's job
            // to release ptable.lock and then reacquire it
            // before jumping back to us.
            proc = p;
            switchuvm(p);
            p->state = RUNNING;
            switch(&cpu->scheduler, p->context);
            switchkvm();
            // Process is done running for now.
            // It should have changed its p->state before coming back.
            proc = 0;
        }
    }
    release(&ptable.lock);
}
```
Xv6 Entering Scheduler

• sched() in proc.c

```c
void sched(void)
{
    int intena;

    if(!holding(&ptable.lock))
        panic("sched ptable.lock");
    if(cpu->ncli != 1)
        panic("sched locks");
    if(proc->state == RUNNING)
        panic("sched running");
    if(readeflags()&FL_IF)
        panic("sched interruptible");
    intena = cpu->intena;
    swtch(&proc->context, cpu->scheduler);
    cpu->intena = intena;
}
```
Xv6 Entering Scheduler (Cont’d)

• When?
  1. Exiting process (exit() in proc.c)

```c
222    // Jump into the scheduler, never to return.
223    proc->state = ZOMBIE;
224    sched();
225    panic("zombie exit");
```

  2. Sleeping process (sleep() in proc.c)

```c
390    // Go to sleep.
391    proc->chan = chan;
392    proc->state = SLEEPING;
393    sched();
```
Xv6 Entering Scheduler (Cont’d)

• When?
  3. Yielding CPU due to timer interrupt
     • trap() in trap.c

```c
if (proc && proc->state == RUNNING && tf->trapno == T_IRQ0+IRQ_TIMER)
    yield();
```

• Yield() in proc.c

```c
acquire(&ptable.lock);  //DOC: yieldlock
proc->state = RUNNING;
sched();
release(&ptable.lock);
```
Project 4 - Priority Scheduler

• Implement priority-based scheduler on xv6
  • The lower nice value, the higher priority
  • The highest priority process is selected for next running
    • Tiebreak: round-robin fashion

• Entering scheduler when
  1. Exiting process
  2. Sleeping process
  3. Yielding CPU
  4. Changing priority
Project 4 - Priority Scheduler

• You also have to make 2 system calls that you did in Project 1. (Parameters, return values, etc. are same with previous project)
  • getnice
  • setnice
    • This time, you will also have to consider entering the scheduler

• When you call fork(), the child process has the same priority as the parent process.

• You don’t need to make or consider thread.
Template Code

• wget http://csl.skku.edu/uploads/SWE3004S19/xv6-project-4.tar.gz

• Modifications
  • halt system call
  • Halt xv6 program
  • make tarball
    • Compress your source codes into one .tar.gz file for submission
    • You should enter your ID & project no. on Makefile
  • CPUS=1
    • Ignore to yield CPU on clock tick
    • yield() system call
Submission

• You need to submit a document.

• Just write how you implemented your code.

• You can use English or Korean.
Submission

• Send your code file (xv6-project-4-studentID.tar.gz) and document file to ks77sj@gmail.com

• Please send a mail with title including [SWE3004-P4]
  • Ex) [SWE3004-P4] 2014111111-project4

• PLEASE DO NOT COPY
  • YOU WILL GET F GRADE IF YOU COPIED

• Due date: 5/8(Wed.), 23:59:59 PM
  • Delays are allowed only one week from the deadline. And there will be up to -40% penalty.
Questions

• If you have questions, please email to TA
  • From this project, you can't ask questions on deadline day

• You can also visit #85533. Please email TA before visiting