Operating Systems
Assignment 2

Krunal Shah (2015EE10476)

1 Add and print priority of the process

1.1 Modify sys_ps to print priority

1. `int priority` parameter was added in the `struct proc` in the file `proc.h`
2. The function `sys_ps` was modified in the file `proc.c` to print the priority of the processes

1.2 Add sys_setpriority system call

The default value of priority for the processes was set as follows:

1. `p->priority = 5` was added in the `found` label in the function `allocproc`

The system call set_setpriority was created as follows:

1. The function `sys_setpriority` is implemented in the file `proc.c` which acquires the lock `ptable.lock`, iterates through the processes in the table and finds the process which matches the pid passed as the argument to the system call. If the state of the found process is any of {`SLEEPING, RUNNABLE, RUNNING`}, it sets the priority of the process to the priority passed to the system call if the priority passed as the argument is less than equal to 20 else it returns -1
2. `#define SYS_setpriority 25` is added to the file `syscall.h`
3. `[SYS_setpriority] sys_setpriority` is added to the array of function pointers `syscalls`
4. `SYSCALL(setpriority)` is added to the file `usys.S`
5. `int setpriority(int, int)` is added to the file `user.h` ther
6. `sys_setpriority` entries are added to `num_calls` and `call_name` arrays for trace printing of the system call

2 Priority Scheduler

Changes were made in the function `scheduler()`. At the beginning of every scheduling, maximum priority of any runnable process in the process table is calculated and then we keep on iterating through the table till we find a process with it's priority equal to the maximum priority and schedule that process.

3 Starvation

3.1 Add sys_getpriority system call

The system call set_getpriority was created as follows:

1. The function `sys_getpriority` is implemented in the file `proc.c` which acquires the lock `ptable.lock`, iterates through the processes in the table and finds the process which matches the pid passed as the argument to the system call and returns it's priority.
2. `#define SYS_getpriority 26` is added to the file `syscall.h`
3. `[SYS_getpriority] sys_getpriority` is added to the array of function pointers `syscalls`
4. `SYSCALL(getpriority)` is added to the file `usys.S`
5. `int getpriority(int)` is added to the file `user.h` ther
6. `sys_getpriority` entries are added to `num_calls` and `call_name` arrays for trace printing of the system call

3.2 Handle starvation

The default value of counter for the processes was set as follows:

1. `int counter` parameter was added in the `struct proc` in the file `proc.h`
2. `p->counter = 0` was added in the `found` label in the function `allocproc`

Changes were made in the function `scheduler()` after the function call `switchkvm()`. At the end of every process execution, we iterate through the RUNNABLE processes in the process table and increment their counter. If the counter becomes 50, we reset it to 0 and increment the priority of the process if it's less than 20.