Operating Systems
Project 3
xv6 Scheduler

Omar Sherif 25-1926
Hazem Amin 22-0542

Files Changed

1. **syscall.h** - define new system calls.

2. **sysproc.c** -

   - write the system call method *sys_settickets(int)* that calls settickets in proc.c.
   - write the system call method *sys_getpinfo(struct pstat*)* that calls getpinfo in proc.c.

3. **syscall.c** -

   - added the external method *sys_getpinfo(void)*.
   - add *sys_getpinfo(void)* to the *syscalls* array.
   - added the external method *sys_settickets(void)*.
   - add *sys_settickets(void)* to the *syscalls* array.

4. **proc.h** added new attributes *tickets,highlow,htickets,ltickets* to the proc structure.

5. **proc.c**

   - manipulated the scheduler function to work as follows :

Scheduling

```
for ever do
    foreach proccess $p \in ptable$ do
        if p.priority=high then
            push p in array highs;
        end
        else
            push p in array lows
        end
        if highs has one element e then
            set e.priority to low;
            run e for one time slice;
        end
        else if highs has more than one element then
            create array ticketholders;
            push in the array the indices of the proccesses in high a
            number of times equal to their tickets;
            generate a random number between 0 and the total
            amount of tickets the highs have;
            get the element $i$ corresponding to that random number in
            ticketholders;
            get the proccess $p$ corresponding to $i$ in highs;
            change $p$'s priority to low;
            run $p$ for one time slice;
        end
        else if lows has one element e then
            run e for two time slices
        end
        else if low has more than one element then
            create array ticketholders;
            push in the array the indices of the proccesses in low a
            number of times equal to their tickets;
            generate a random number between 0 and the total
            amount of tickets the lows have;
            get the element $i$ corresponding to that random number in
            ticketholders;
            get the proccess $p$ corresponding to $i$ in lows;
            run $p$ for two time slices;
        end
    end
end
```

   - write the system call method *sys_settickets(int)* that changes the number of tickets of the currently running proccess.
   - write the system call method *sys_getpinfo(struct pstat*)* that fills a pstat structure with proccess information from the proccess table.

6. **user.h** - add syscalls definition.
7. **usys.S** -
   - define *getpinfo* as a system call.
   - define *settickets* as a system call.
8. **customps.c** - created for the purpose of calling the system call getpinfo and printing the results to the console.
9. **schtest.c** - created for the purpose of testing the scheduler,it forks three child proccesses and they are tracked to see the frequency in which they get cpu time.
10. **Makefile** -
    - add *customps* to the list of user programs.
    - add *schtest* to the list of user programs.
    - changed the number of cpus to one.
11. **pstat.h** added for adding the structure pstat.

Statistics

using 3 processes in schtest.c here is the gant chart for different tickets given to each
P1:10 **P2**:15000 **P3**:700

P2	P1	P2	P3	P2	P3	P1

P1:50 **P2**:150 **P3**:50

P3	P2	P3	P1	P3	P2	P1	P3

bitbucket account: osherifo