Xv6 Assignment

REPORT

Roll no : 20161170
Name : Shubh

Original Xv6 Scheduler

The original Xv6 Scheduler has round robin scheduler.

(The test program is given on last page)

On testing the original scheduler with three instances of the “foo” program, it was observed that the process which was started earlier finished earlier and there is no way to set priority or make a runnable program running unless a program exits.

Priority based Xv6 Scheduler

The modified Xv6 Scheduler has priority based scheduler.

(The test program is given on last page)

On testing the original scheduler with three instances of the “foo” program, it was observed that priority of a process can be changed and we can make a runnable program running even before the running program exits.

Proof is given on next page.

(Order of starting time : 14 < 7 < 18)

```
$ ps
name     pid     state           priority
init     1       SLEEPING        60
sh       2       SLEEPING        60
ps       19      RUNNING         60
foo      14      RUNNABLE        60
foo      7       RUNNABLE        60
foo      18      RUNNING         60
```

(Increasing the priority of pid 7)

```
$ set_priority 7 0
 pid=7, pr=0
Previous priority of     PID 7 is 60
Current priority of      PID 7 is 0
```

(List of processes)

```
$ ps
name     pid     state           priority
init     1       SLEEPING        60
sh       2       SLEEPING        60
ps       25      RUNNING         60
foo      14      RUNNABLE        60
foo      7       RUNNING         0
foo      18      RUNNABLE        60
```

(List of processes)

```
$ ps
name     pid     state           priority
init     1       SLEEPING        60
sh       2       SLEEPING        60
ps       25      RUNNING         60
foo      14      RUNNABLE        60
foo      18      RUNNABLE        60
```

(foo with pid 7 exited before pid 14, 18)

Test Program (foo.c)

```c
#include "types.h"
#include "stat.h"
#include "user.h"
#include "fcntl.h"

int
main(int argc, char *argv[])
{
  int id = fork();
  if (id == 0) {
    double z, d = 1, x = 0;
    for (int i = 0; i < 5; ++i) {
      for (int j = 0; j < 10; ++j) {
        for (z = 0; z < 8000000; z += d) {
          x = x + 3.14 * 89.64; // useless calcs
        }
      }
    }
    printf(1, "foo ended\n");
  }
  exit();
}
```