Online on xv6 - Scheduler

Section: A2 Time: 30 minutes

Your task is to implement a simple priority based scheduler. You are given a user program testloop.c, which simply simulates a long running job by iterating a loop a given number of times. This program takes two arguments: iteration count and priority. You will also need to implement the system call to set priority for each process.
The process with the highest priority will keep running till completion, unless another process with higher priority arrives. In that case, the higher priority process will run till completion. **All processes have a default priority of 300**. **Processes with the same priority will run in a RR fashion.**

Input:

```
testloop 200 10 &;
testloop 150 8 &;
testloop 100 15 &;
testloop 90 8 &;
```

Output:

```
PID 5 (10): Starting 200 iterations at time 35
PID 8 (15): Starting 100 iterations at time 54
PID 8 (15): Finished at time 175
PID 5 (10): Finished at time 265
PID 11 (8): Starting 150 iterations at time 265
PID 12 (8): Starting 90 iterations at time 266
PID 12 (8): Finished at time 370
PID 11 (8): Finished at time 370
```

Note:
Set CPUS := 1 in the Makefile. You must provide the input in the shell in separate lines (not all at once). Be quick to provide the inputs, otherwise your ordering may not match with the given one. Increase the iteration count if needed.

Notice how the priorities change the execution order of each process. At first, only the process with priority 10 is present, so it keeps running. Then comes a process with a lower priority 8, so it does not affect the running process. But the next one has higher priority (15), so it preempts the current process and runs to completion. And finally another process with priority 8 arrives, it runs in RR schedule alongside the process with PID 11.

Submission:

```
git add --all
git diff HEAD > ../2005010.patch
```