Online on xv6 - Scheduler

Section: A1 Time: 30 minutes

Your task is to implement the preemptive SJF scheduling algorithm in xv6. You are provided a user program `testloop.c`. It simulates a long running task by running a for loop. The iteration count is provided as a single argument. Examine its contents to understand how it works.

The **default length** for each job is **10**. The job length for the testloop program is its iteration count. You may add fields in the proc structure for keeping track of the job lengths.

Input:

```
testloop 120 &;
testloop 110 &;
testloop 100 &;
ls
```

Output:

```
Process 5: Starting 120 iterations at time 35
Process 8: Starting 110 iterations at time 70
Process 11: Starting 100 iterations at time 97
<output of ls, omitted for brevity>
Process 11: Finished at time 167
Process 8: Finished at time 216
Process 5: Finished at time 265
```

Note:

Set CPUS := 1 in the Makefile. You must provide the input in the shell one by one (not all at once). Be quick to provide the inputs, otherwise your ordering may not match with the given one. The colors are present only for tracking each process.

Notice how the ordering is present: whenever a shorter job arrives, the scheduler will keep running that job till completion. Since `ls` is the shortest job, it immediately starts running and completes its execution. After that, the next shortest job (PID=11) completes, and then the next one (PID=8), and so on.

Submission:

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