System Call Implementation

Implement the system call pstate in xv6. Similar to the Unix command ps, it prints out the state of the current user processes. For simplicity, there is no argument to the command.

- Consider only those user processes in SLEEPING, RUNNING, or RUNNABLE state.
- The display for each process should include process id, process name, process state, and parent name.
- Use *(init)* for the parent name of the *init* process.
- The command also prints out the total number of the processes that are currently in SLEEPING, RUNNING, or RUNNABLE state.
- The command also prints out the status of each CPU, where the status is either *idle* or the name of the current process it is running.
- sh.c should not be modified.
- There will be a penalty for any unnecessary modifications to xv6.
- The implementation of additional user programs (like the pi in the sample run) for testing purpose is part of the assignment.

Sample run:

```
$ pi &
$ pi &
$ pi &
$ pi &
$ pi &
$ pstate
pid     name    state           parent
-------------------------------------
1       init    SLEEPING        (init)
2       sh      SLEEPING        init
13      pstate  RUNNING         sh
4       pi      RUNNABLE        init
6       pi      RUNNING         init
8       pi      RUNNABLE        init
10      pi      RUNNABLE        init
12      pi      RUNNING         init
total: 8
cpu 0: pstate
cpu 1: pi
cpu 2: pi

$ pstate
pid     name    state           parent
-------------------------------------
1       init    SLEEPING        (init)
2       sh      SLEEPING        init
14      pstate  RUNNING         sh
4       pi      RUNNABLE        init
6       pi      RUNNABLE        init
8       pi      RUNNING         init
10      pi      RUNNING         init
12      pi      RUNNABLE        init
total: 8
cpu 0: pi
cpu 1: pi
cpu 2: pstate

$
```

Submission: a zipped file firstname_lastname.zip consisting of all modified xv6 files (including Makefile) and all new files (including at least one file for testing).