

Changes in scheduler:

The pre built scheduler of xv6 is round robin. The scheduler you are going to implement is based on lottery scheduling and MLFQ. This require adding something new to proc structure found in proc.h to accommodate number of tickets associated with each process as well as the queue on which process is running currently. After adding this you will also need to initialize them whenever a new process is allocated meaning you need to do some initialization of variables added previously in proc struct.

Implementing Scheduling Policy:

1. Start with topmost priority queue.
2. If there are more than one processes on that queue determine winning process else move to the next queue.
3. To determine winning process follow these steps:
 - a. First count total number of tickets associated with particular process on that queue.
 - b. Generate a random number out of this count.
 - c. Determine winning process by finding a process where ticket number sum upto this point is greater then this count.
4. After determining winning process update any states that require to be updated.
5. Consider giving two time slices to a process on low level queue. To do this you may need to consider yield() function.

Changes to be made:

- In proc.h
- In proc.c
- Implementing two new system calls
- Making a test file at user side that will help you to draw graph
- Including pstat.h

Making test function:

To test your scheduler create more than one process in your function and then assign different number of tickets to each created process. Then get information about running process using getpinfo. Both these tasks will be performed using system calls named as getpinfo and settickets.

Making Graph:

Make a graph out of the results obtained from your test function. You should explain relation between the tickets assigned and proportion of CPU cycles that is given to each process.