

CS39002: Operating Systems Lab
Spring 2017

Assignment 5
Due: March 26th, 6 pm

In this assignment you will be working with the Pintos OS that was discussed in class. Specifically, your job in this assignment is to:

1. Installing Pintos on your laptop/PC, and running user programs on it
2. Understand the source codes, especially the files `thread.c`, `pallocc.c`, `malloc.c`, `sync.c` in the `src/threads` directory, and any other files that functions in these files call. Also take a look at the `src/lib` directory. You should try to understand how the thread management (data structures kept, creation, scheduling, blocking/unblocking, context switch, termination) and memory management work in Pintos.
3. Write code to replace the default Pintos scheduler with the following scheduler:
 - a. A multilevel feedback queue with two levels, Level 1 and 2, that works as follows. Let the time quanta in the default round robin scheduler in Pintos be T .
 - i. Threads are added on creation to Level 1 queue.
 - ii. Level 2 queue threads are scheduled only if Level 1 queue is empty.
 - iii. Level 1 scheduling policy is round-robin with time quanta equal to T . However, if a thread runs for two time quanta and still does not finish, it is pushed down to the end of the Level 2 queue.
 - iv. Level 2 scheduling is round-robin with time quanta equal to $2T$. However, if a thread in Level 2 waits for $6T$ and still does not get the CPU (because Level 1 queue is not empty), it is pushed up to the end of the Level 1 queue.
4. Write code to replace the default implementations of `malloc`/`realloc`/`free` in Pintos with a memory management system that uses the **buddy memory allocation algorithm** (read from the text). Use the buddy algorithm to allocate memory inside each page, getting a new page from the page allocator if sufficient memory is not available in the existing pages.

Submit the following:

1. A design document (nicely formatted) explaining (i) how the existing Pintos Scheduler works (the scheduling policy is simple, explain how a context switch happens stating the functions called in sequence and what does each function do (ii) data structures you have added/modified (iii) which functions

you have added/modified (for added function (if any), explain what it does; for modified functions explain what you modified) (iv) which files you have modified.

2. A design document with the same things as above for the memory management part. Explain the within-page memory management, not the page allocator).
3. **ONLY** the files that you have changed in Pintos. You must **NOT** add any new files. Pintos should build with just the default files replaced with your files. Minimize the number of files that you have to change to do steps 3 and 4.