Apollo Truong
861168740
Sabrina Wong
861195047

Design document: Lab #2

In this lab we implemented a setpriority function that will change the priority of a process. The highest priority will be 31, while the lowest priority will be 0.

The scheduler will run and enable interrupts. We added a segment of code to scan the ptable for the process with the highest priority. When this highest priority is found, it will then run acquire(&ptable.lock), then run the process. The scheduler will then call release(&ptable.lock) to unlock the process table. The scheduler will then loop over constantly.

The program may be rewritten to make 0 the highest priority.

The headings

The files we changed:
Proc.c (majority) inside of scheduler() function

```
for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
       if( p->state != RUNNABLE){
         continue;
       }
       if(hi_prty < p->priority){ // reset the max priority to new value;
         hi_prty= p->priority;
       }
     }
```

defs.h (124)

```
void        setpriority(int);
```

syscall.c (105,132)

```
extern int sys_setpriority(void);
```

```
[SYS_setpriority] sys_setpriority,
```

syscall.h (24)

```
#define SYS_setpriority 23
```

sysproc.c (50)

```
#define SYS_setpriority 23
```

user.h (27)

```
int setpriority(int);
```

usys.S (34)

SYSCALL(setpriority)