Part 2 - xv6 Extension

1. Add `priority` attr to proc in `proc.h`;

```
// Per-process state
struct proc {
  uint sz;                     // Size of process memory (bytes)
  pde_t* pgdir;                // Page table
  char *kstack;                // Bottom of kernel stack for this pro
  enum procstate state;        // Process state
  int pid;                     // Process ID
  struct proc *parent;         // Parent process
  struct trapframe *tf;        // Trap frame for current syscall
  struct context *context;     // swtch() here to run process
  void *chan;                  // If non-zero, sleeping on chan
  int killed;                  // If non-zero, have been killed
  struct file *ofile[NOFILE];  // Open files
  struct inode *cwd;           // Current directory
  char name[16];               // Process name (debugging)
  int priority; // this is for the process priority
};
```

2. Add iteration through priority in scheduler, 3 priority levels range from 0 to 2;

```
for(int priority = 0; priority <= 2; priority++) {
  for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
    if(p->state != RUNNABLE || p->priority != priority)
      continue;
```

3. implement renice(), renice should take pid to recognize process, and the priority it expected as a second paremeter, we also need to add a wrapper for renice(), as sys_renice() should serve as a sys call;

```c
int
renice(int pid, int priority)
{
  struct proc *p;
  int flag = 0;

  // lock critical section
  acquire(&ptable.lock);
  // if the proc exists, then stop iterate
  for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
    if(p->pid == pid) {
      p->priority = priority;
      flag = 1;
      break;
    }
  }
  release(&ptable.lock);

  if(flag)
    return 0;
  else
    return -1;
}
```

```c
int
sys_renice(void)
{
  int pid, priority;

  if(argint(0, &pid) < 0 || argint(1, &priority) < 0)
    return -1;

  return renice(pid, priority);
}
```

4. add syscall signature in `syscall.c`, `def.h`, and `syscall.h`

```
3    [SYS_ps]      sys_ps,
4    [SYS_renice]  sys_renice,
5
6    };
```

```
24   #define SYS_ps      23
25   #define SYS_renice 24
26
```

```
120  int           wait(void);
121  void          wakeup(void*);
122  void          yield(void);
123  int           ps(void);
124  int           renice(int, int);
125
```

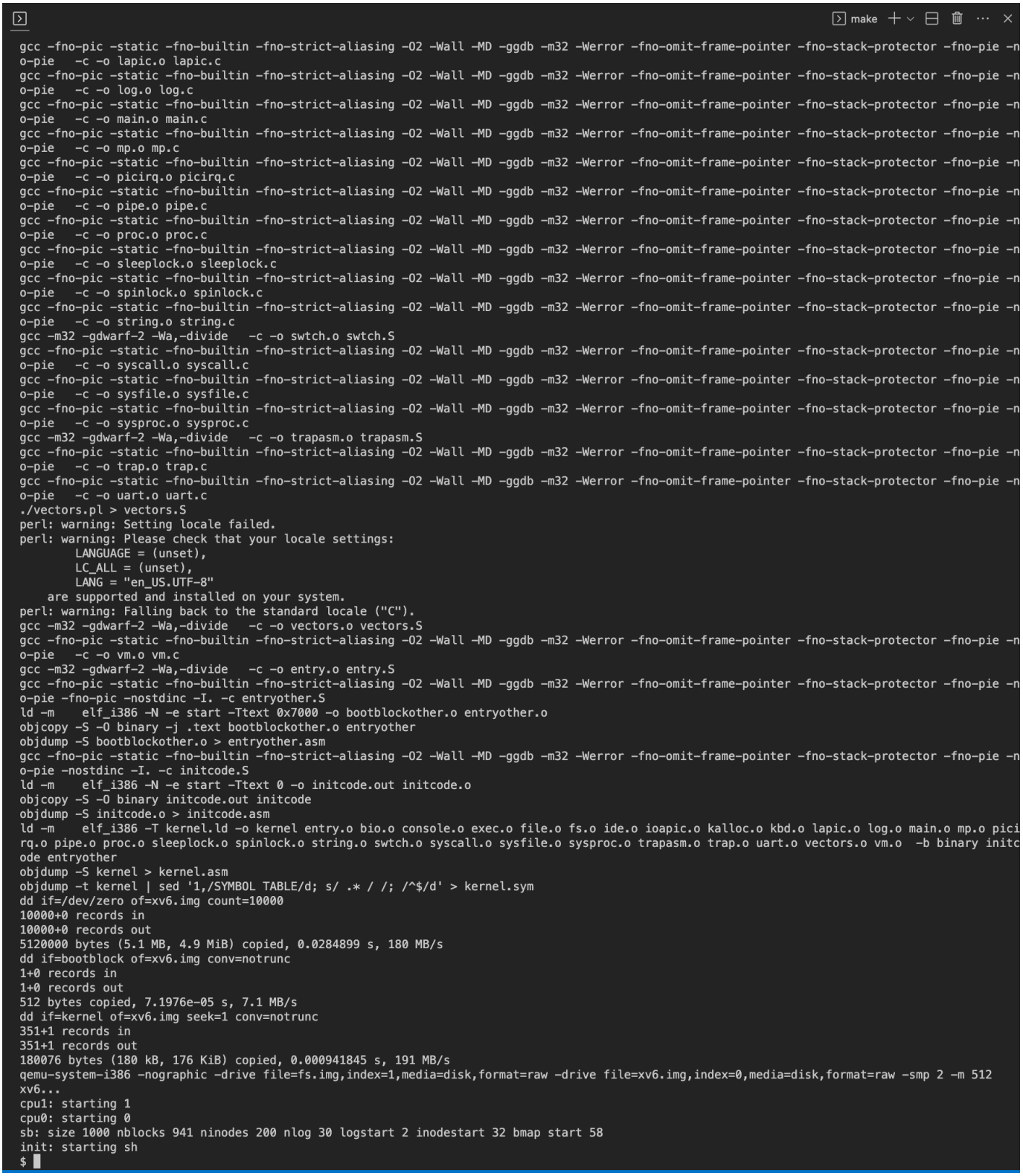
5. add a test for the renice() syscall, we use fork to generate new child processes, and set new priority to them based on RR, print the info for process priority and iteration time after the `renice()` promotion;

```c
#include "types.h"
#include "user.h"
#include "stat.h"

int
main(int argc, char *argv[])
{
  int pid;
  int priority;

  for (int i = 0; i < 5; i++) {
    pid = fork();
    if (pid < 0) {
      printf(1, "ERROR: fork failed\n");
      exit();
    }
    else if (pid == 0) {
      priority = (i % 3); // give priority by iteration RR
      renice(getpid(), priority); // set priority via renice
      for (int j = 0; j < 50; j++) {
        printf(1, "Process %d with priority %d is running, iteration %d\n", getpid(), priority, j);
        sleep(10); // Sleep to switch to other processes
      }
      exit();
    }
  }

  // Parent wait
  for (int i = 0; i < 5; i++) {
    wait(); // join one thread
  }

  exit();
}
```

6. Makefile change (add _mlfq);

```
168  UPROGS=\
169      _cat\
170      _echo\
171      _forktest\
172      _grep\
173      _init\
174      _kill\
175      _ln\
176      _ls\
177      _mkdir\
178      _rm\
179      _sh\
180      _stressfs\
181      _usertests\
182      _wc\
183      _zombie\
184      _hello\
185      _date\
186      _ps\
187      _mlfq\
```

7. Screen capture for compile and exec result;


make
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o lapic.o lapic.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o log.o log.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o main.o main.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o mp.o mp.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o picirq.o picirq.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o pipe.o pipe.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o proc.o proc.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o sleeplock.o sleeplock.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o spinlock.o spinlock.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o string.o string.c
gcc -m32 -gdwarf-2 -Wa,-divide   -c -o swtch.o swtch.S
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o syscall.o syscall.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o sysfile.o sysfile.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o sysproc.o sysproc.c
gcc -m32 -gdwarf-2 -Wa,-divide   -c -o trapasm.o trapasm.S
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o trap.o trap.c
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o uart.o uart.c
./vectors.pl > vectors.S
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = (unset),
        LC_ALL = (unset),
        LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
gcc -m32 -gdwarf-2 -Wa,-divide   -c -o vectors.o vectors.S
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie   -c -o vm.o vm.c
gcc -m32 -gdwarf-2 -Wa,-divide   -c -o entry.o entry.S
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie -fno-pic -nostdinc -I. -c entryother.S
ld -m   elf_i386 -N -e start -Ttext 0x7000 -o bootblockother.o entryother.o
objcopy -S -O binary -j .text bootblockother.o entryother
objdump -S bootblockother.o > entryother.asm
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -n
o-pie -nostdinc -I. -c initcode.S
ld -m   elf_i386 -N -e start -Ttext 0 -o initcode.out initcode.o
objcopy -S -O binary initcode.out initcode
objdump -S initcode.o > initcode.asm
ld -m   elf_i386 -T kernel.ld -o kernel entry.o bio.o console.o exec.o file.o fs.o ide.o ioapic.o kalloc.o kbd.o lapic.o log.o main.o mp.o pici
rq.o pipe.o proc.o sleeplock.o spinlock.o string.o swtch.o syscall.o sysfile.o sysproc.o trapasm.o trap.o uart.o vectors.o vm.o -b binary initc
ode entryother
objdump -S kernel > kernel.asm
objdump -t kernel | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > kernel.sym
dd if=/dev/zero of=xv6.img count=10000
10000+0 records in
10000+0 records out
5120000 bytes (5.1 MB, 4.9 MiB) copied, 0.0284899 s, 180 MB/s
dd if=bootblock of=xv6.img conv=notrunc
1+0 records in
1+0 records out
512 bytes copied, 7.1976e-05 s, 7.1 MB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
351+1 records in
351+1 records out
180076 bytes (180 kB, 176 KiB) copied, 0.000941845 s, 191 MB/s
qemu-system-i386 -nographic -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$

```
$ $
$ mlfq
Process 12 with priority 0 Process 13 with priority 1 is running, iteration 0
Process 15 with priority 0 is running, iteration 0
Process 14 with priority 2 is running, iteratioProcess 16 with priority 1 is running, iteration 0
is running, iteration 0
n 0
ProcPess 15 with priority 0 is rPPrroorocess 13 with priority 1 is running, iteration 1
ceunning, iteration 1
ss 12 with priority 0 is running, iteration 1
Prcess 16 with priority 1 is running, iteration 1
ocess 14 with priority 2 is running, iteration 1
Process 13 with priority 1 is running, iteration 2
Process 12 with priority 0 is running, iteration 2
Process 15 with priority 0 is running, iteration 2
Process 16Process 14 with priority 2 is running, iteration with priority 1 is running 2
Process 13 with priority 1 is running, iteration 3
, iteration 2
Process 12 with priority 0 is running, iteration 3
Process 15 with priority 0 is running, iteration 3
ProProPProcess 12 with priority 0 is running, iteration 4
cess 13 with priority 1 is running, rocProecess 15 with priority 0 is running, iitercess 16 with priority 1 is running, iteration 3
ss 14 with priority 2 is running, iteration 3
teration 4
ation 4
Process 12 with priority 0 is running, iteration 5
ProcesPrs 16 with priority 1 is running, iteration 4
Process 14 with priority 2 is runnocess 15 with priority 0 is running, iteration 5
ing, iteration 4
Process 13 with priority 1 is running, iteration 5
Process 12 with priority 0 is running, iteration 6
PrProcess 16 with priority 1 is running, iteration 5
Process 14 with priority 2 is ocess 15 with priority 0 is running, iteration 6
running, iteration 5
Process 13 with priority 1 is running, iteration 6
Process 12 with priority 0 is running, iteration 7
ProProPrceocesss 16 with priority 1 is running, ics 15 with priority 0 is running, iteration 7
teration 6
Process 14 with priority 2 iess running, iteration 6
Process 12 with priority 0 is running, iteration 8
s 13 with priority 1 is running, iteration 7
PrProcess 15 with priority 0 is running, iteration 8
Process 14 with priority 2 is running, iteratiocess 16 with priority 1 is running, iteration 7
on 7
Process 12 with priority 0 is running, iteration 9
Process 13 with priority 1 is running, iteration 8
Process 15 with priority 0 is running, iteration 9
Process 16 with priority 1 is running, iteration 8
Process 14 with priority 2 is runningProc,Pess 12 with priority 0 is running, iteration 10
Process 15 with priority 0 is running, iteration 10
rocess 13 with priority 1 is running, iteration 9
```