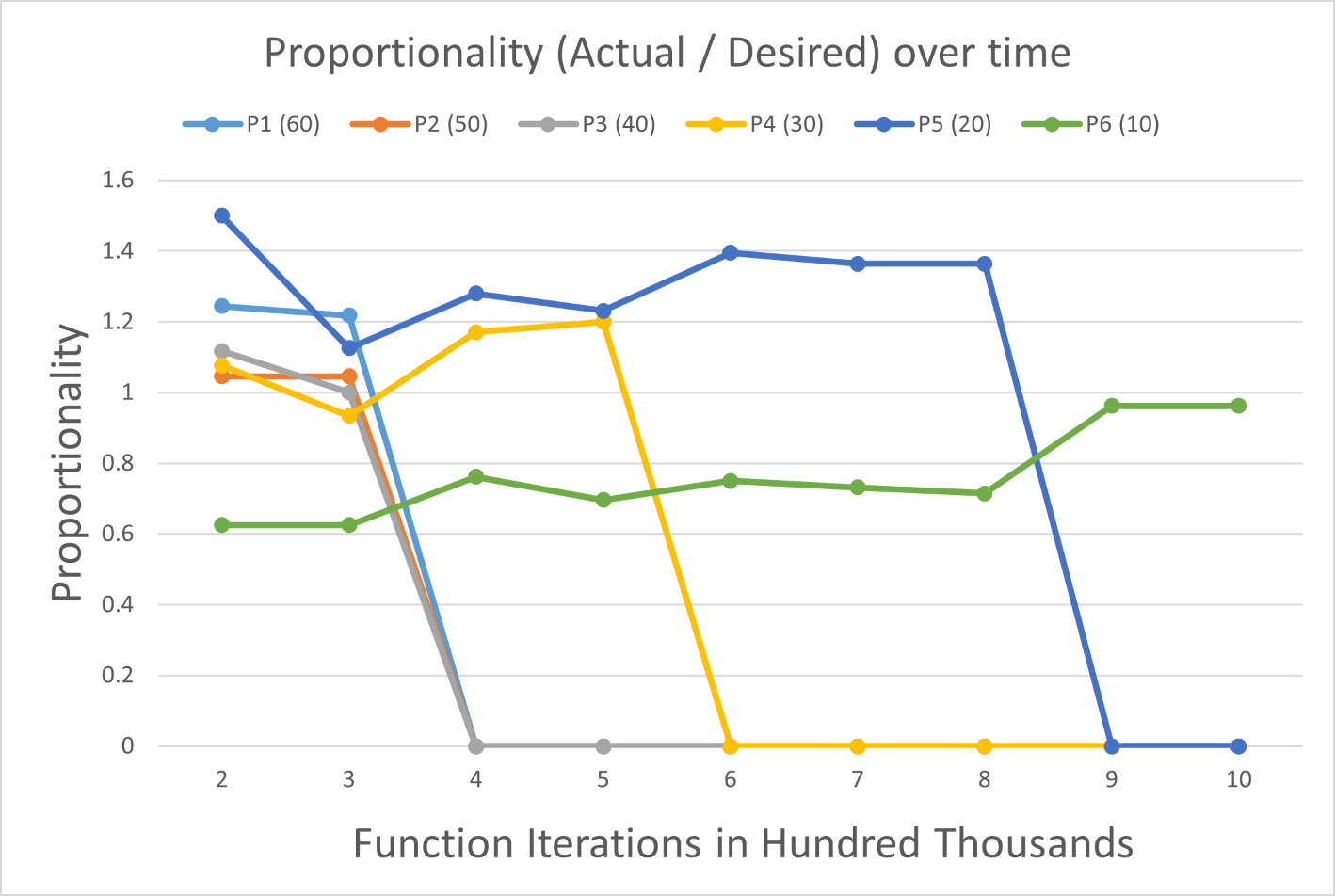
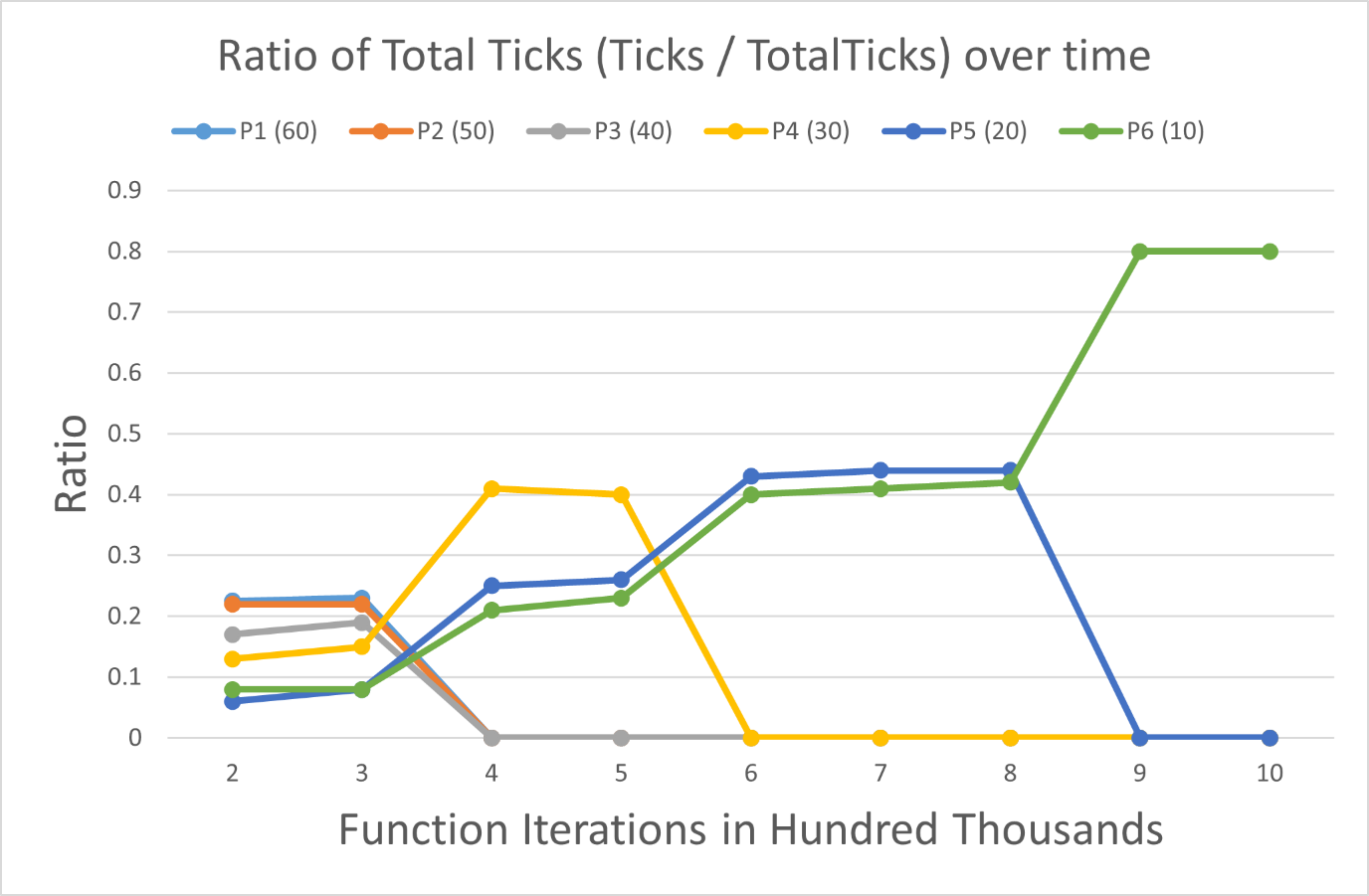
Project 3: Lottery Scheduling

Gavin Osborn | CSCI-320 | 10/20/23

The goal of project 3 was to implement lottery scheduling in xv6 RISC-V. This entailed creating two new syscalls (settickets and getpinfo). In addition to the scheduler, a command-line program *ps* was to be created alongside a user test file*, lotteryTest*.

I have completed all of these requirements, and the following graphs indicate all to be working as intended. Besides these two graphs, I have also included a list of changed files.  
  


**Fig 1. Proportionality over time.** Trend in remaining proportionality to desired scheduling ratio over time for given processes. Test file created six processes, with tickets allotted in respective increments of ten. Data collected every one hundred thousand iterations of a work-loop.



**Fig 2. Ratio of Total Ticks over time.** Scheduling ratio over time for given processes.Test file created six processes, with tickets allotted in respective increments of ten. Data collected every one hundred thousand iterations of a work-loop.

**Changed Files**

**\*kernel/param.h\***

#define DEFAULT\_TICKETS 1 // default number of tickets

#define DEFAULT\_SEED 20 // default seed for RNG

**\*kernel/syscall.h\***

#define SYS\_settickets 22

#define SYS\_getpinfo 23

#define SYS\_rand\_init 24

#define SYS\_scaled\_random 25

**\*kernel/syscall.c\***

extern uint64 sys\_settickets(void);

extern uint64 sys\_getpinfo(void);

extern uint64 sys\_rand\_init(void);

extern uint64 sys\_scaled\_random(void);

[SYS\_settickets] sys\_settickets,

[SYS\_getpinfo] sys\_getpinfo,

[SYS\_rand\_init] sys\_rand\_init,

[SYS\_scaled\_random] sys\_scaled\_random,

**\*kernel/sysproc.c\***

defined sys\_getpinfo() and sys\_settickets()

defined sys\_rand\_init() and sys\_scaled\_random()

**\*kernel/proc.h\***

added tickets field

added ticks field

added extern struct proc proc[NPROC];

**\*kernel/proc.c\***

fork()

-> set # of tickets to same as parent

allocproc()

-> set default tickets to 1

Scheduler()  
-> implemented lottery scheduling

getpinfo(struct pstat \*pFromUser);

get\_total\_tickets(void);

**\*user/user.h\***

int settickets(int);

int getpinfo(struct pstat \*);

**\*user/usys.pl\***

entry("getpinfo");

entry("settickets");

entry("rand\_init");

entry("scaled\_random");

**\*kernel/pstat.h\***added

**\*kernel/random.h\***

added

**\*kernel/random.c\***

added

**\*user/lotteryTest.c\***

created

**\*user/ps.c\***

created