Mini Project 1 Fall 2019 (System calls)

Rubric

Report 30%
Presentation 20 %
Code and documentation 50%

Participants name	Code Section	Report Section	Documentation Sec	Presentation Sec

Submission deadline: Sep 24th at 11:59 pm.
Presentation deadline: Sep 24th at class time.

Deliveries:

- Zip file with XV6 already modified
- Report: Description explaining the reason and how to make the kernel modifications. This include screenshots of the modification and screenshots of the output.

Complete list of mini-projects in three easy pieces website, column (Appendices) -> xv6 Labs
http://pages.cs.wisc.edu/~remzi/OSTEP/lab-projects-xv6.pdf

General description

In chapter 6 (Direct Execution) we talked about system calls, and how the OS provides this mechanisms for allowing process in user mode to execute privileged instruction.

In this homework mini-project your team will add a simple system call to xv6. Your new syscall should look like this: int getpidcount(void) Your system call returns the value of a counter (perhaps called pidcount or something like that) which is incremented every time any process calls the getpid() system call. That's it!

Hint

You can start by taking a look and using another system call, like **getpid()** to get an idea of how the code works . Most of the time will be spent on understanding the code. There shouldn't be a whole lot of code added you can use any editor and gcc as a compiler.

Check this Lab tutorial for C programming in three easy pieces website:
http://pages.cs.wisc.edu/~remzi/OSTEP/lab-tutorial.pdf

Details

Setting up the environment.

Download XV6 from Canvas Modules->Resources-XV6
Unzip the file

tar –xzf XV6.tar.gz

Compile the system

Make

Modify file Makefile line 56

QEMU:=qemu-system-i386

Run qemu

Make qemu-nox

More instructions about the homework and details about the different components of xv6
b. https://www.youtube.com/watch?v=lyuNbuymGD8&feature=youtu.be

Excellent video (Prof Remzi Arpaci-Dusseau Author of three easy pieces)
start watching at minute 37:57

Just for this time, this are the files you will have to modify

- syscall.h
- syscall.c
- user.h
- usys.S
- sysproc.c
- Makefile

Resources:

a. Installing XV6 and intro to system calls
https://www.youtube.com/watch?v=vR6z2QGcoo8

b. Some Background about system call in XV6 https://github.com/remzi-arpacidusseau/ostep-projects/blob/master/initial-xv6/background.md

c. https://www.youtube.com/watch?v=lyuNbuymGD8&feature=youtu.be
Excellent video (Prof Remzi Arpaci-Dusseau Author of three easy pieces) start watching at minute 37:57

Check in class how to create the program that use the system call.

Your driver program should looks somehow like that

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

// running system call howmanysys()
int
main(void)
{
    printf(1, "There are %d system calls.\n", getreadcount());
    exit();
}
```