CS 1550

Lab 3 – xv6 Exercise

Teaching Assistant

Henrique Potter

xv6 exercise hints

- We need to worry about two things:
  - How to count syscalls?
  - Implement the method to return the counting of syscalls

- Past slides are available in:
  - http://people.cs.pitt.edu/~henriquepotter/

xv6 exercise hints

- **Syscall** calls will need variable to hold the counting values
  - Where to write this data structure?
    - Which file holds process metadata? **proc.c**
  - Which data structure?
    - Each **syscall** have an id, which could be used as?
    - Which basic data structure uses indexes for element positions?
  - Important method can be found in syscall.c
    - syscall(void) -> Is called every time any syscall is called

xv6 exercise hints

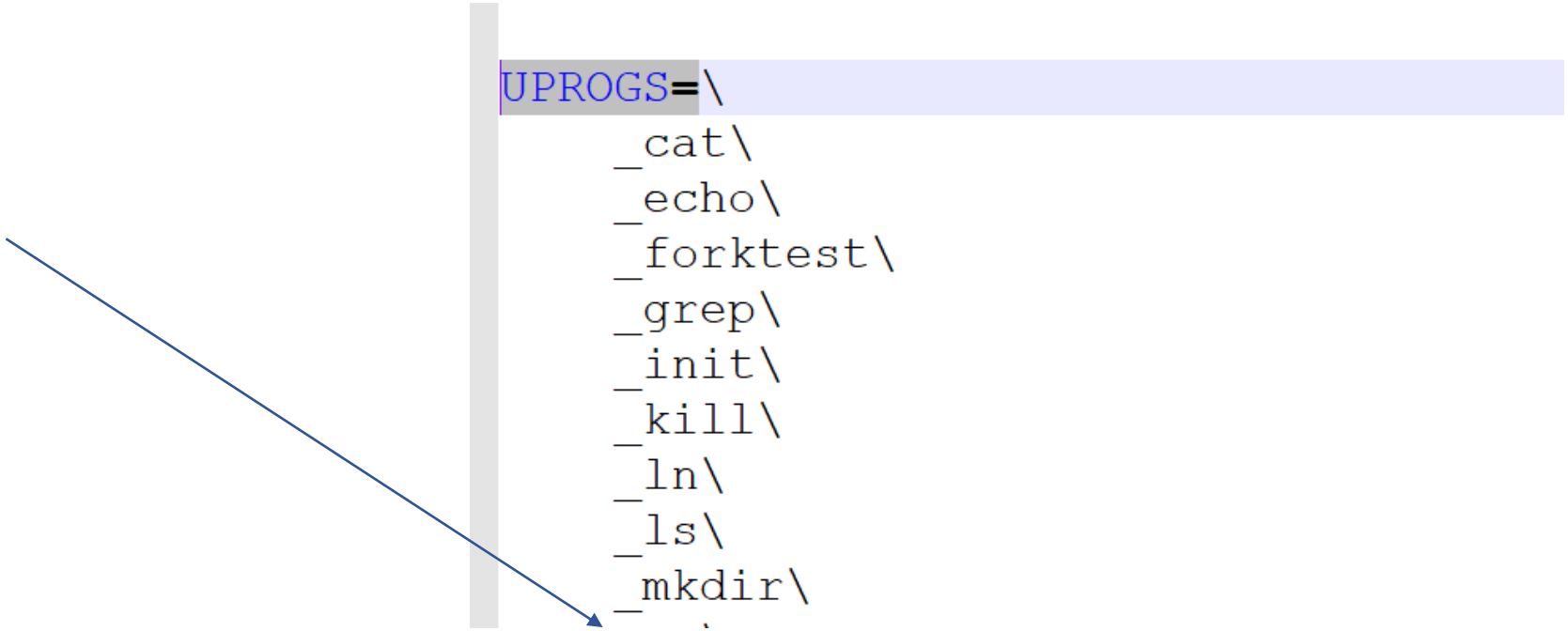
```
void syscall(void){
        int num;
        struct proc *curproc = myproc();
        num = curproc->tf->eax;

        if(num > 0 && num < NELEM(syscalls) && syscalls[num]) {
                curproc->tf->eax = syscalls[num]();
        }
        else {
                cprintf("%d %s: unknown sys call %d\n", curproc->pid, curproc->name, num);
                curproc->tf->eax = -1;
        }
}
```

xv6 exercise hints

- Implementing **getcount**
  - Specify the method and its **id** in **syscall.h**
  - Specify extern method and pointer
    - **syscall.c**
  - Where to implement **int sys_getcount(void)**
    - **sysproc.c**
  - Add SYSCALL(getcount)
    - **usys.S**

xv6 exercise hints

- Adding an user program
  - Open makefile
- Add
  - _getcount\

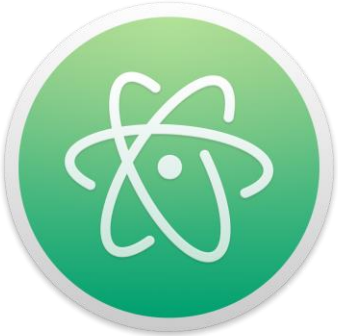


Lab Assignment

- **Due**: Friday, September 21, 2018
- Leave the files in your linux.cs.pitt.edu user account folders **for now**

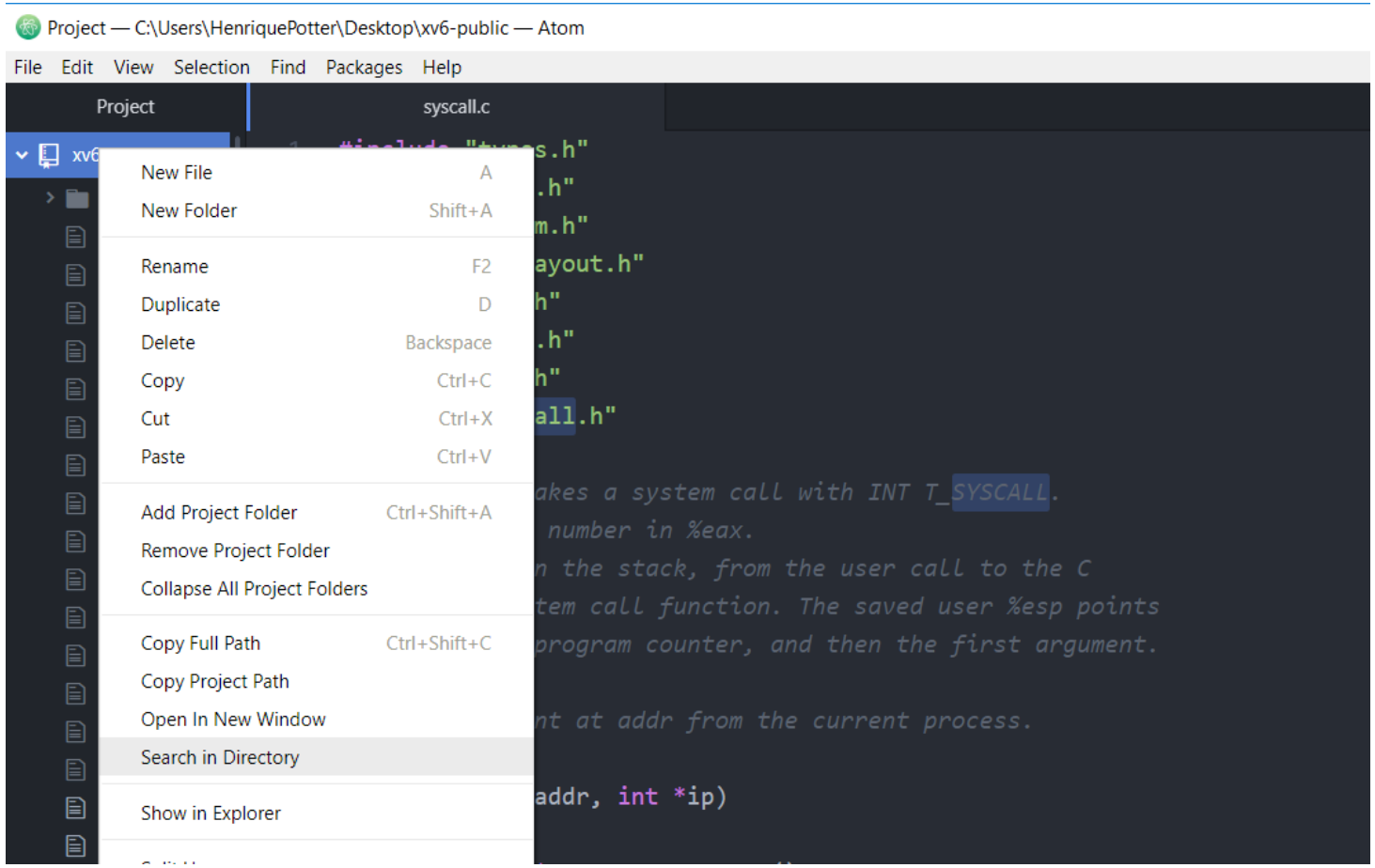
Atom io

- Good coding tool
- Easily search for variables names in any file inside a folder
- Lots of plugins



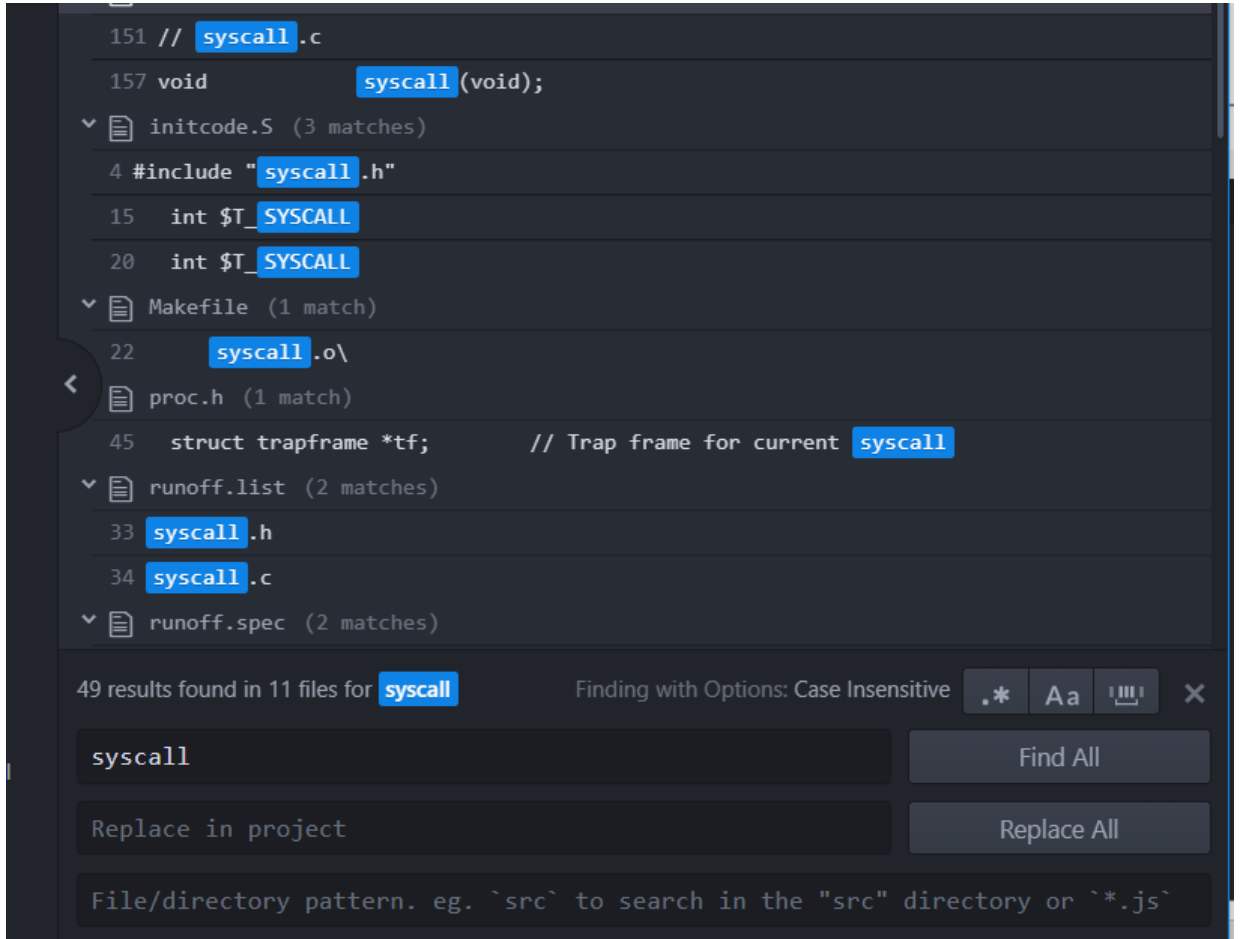
Atom io

- Search for variables in any file from a Directory



Atom io

- Search for variables in any file from a Directory



CS 1550

Lab 3 – xv6 Exercise

Teaching Assistant

Henrique Potter