About the Final Exam

The final exam is scheduled for Thursday, December 14, 13:45 - 15:45 in the same room that we meet. It will be an in-person, on-paper, closed-notes, closed-book exam. The questions on the exam will be based on the lecture notes and the assignments.

Question Types

The question types will be a mix of true/false, multiple choice, fill-in, short-answer, and code-writing questions. Sample question types follow. Unless otherwise indicated, all answers should be based on Linux.

1. What is the POSIX standard data type that represents a login record?
2. What value does the `read()` system call return to indicate the end of a file?
3. What command will display the date of creation of the file `/etc/bash.bashrc`?
4. True or False: A directory can have a link in more than one other directory.
5. A process can obtain its process ID with which system call?
6. What header file must a program include to use the `scanf()` function?
7. Using octal modes only, write a command that will change the permission of the file named foo in the current working directory so that the owner has read, write, and execute permission, the group has read and write permission, and everyone else has read and execute permission.
8. Write a C program that prints out the words on its command line in reverse order, one per line.
9. List all of the different file types in a Unix system.
10. Write a command that will generate a random permutation of the numbers from -99 to 0 in a file named `numbers`, in the current working directory, overwriting it if it already exists.
11. Given the program below, named `/data/biocs/b/cs493.66/prog.c`, suppose it is compiled into an executable named `prog` in the same directory as its source code. What is output if the user runs this program?

```
#include <stdio.h>
#include <string.h>
void main(int argc, char * argv[])
{
    char *ptr, *ptr2 = NULL;
    ptr = strrchr( argv[0], '/');
    if ( ptr != NULL )
        ptr2 = ptr+1;
    else
        ptr2 = argv[0];
    if ( ptr2 != NULL )
        printf("%s\n", ptr2);
}
```