Logan Floyd 4/27/22

ECE282 Spring 2022

Final Exam

April 27th 2022

1) The command **who am i** shows what information and how is it different from the command whoami?

   who am I diplays your username and when you get logged in, while whoami diplays your own username.

2) When reading a file with the command **cat /etc/passwd**, how would you use the **grep** command to retrieve your username information from the file.

   cat /etc/passwd |grep $USERNAME > myname.txt

3) What will the command **rmdir** do? Be specific on what is necessary for it to work removing a directory.
   rmdir – command used for deleting empty directories
   rmdir dirname

4) How do you use the rm command to remove a directory and all its contents? Be specific

   rm -d dirname (for an empty directory)

   rm -r dirname (for a non-empty directory)
   rm -rf dirname (for a non-empty directory without having to be prompted for file deletion)

5) Show an example of using the cp command to copy a file up two directory levels. Hint: recall what the dot (.) and double dot (..) are in a directory listing.

   cp text.txt text2.txt ../..

6) The data type Unix uses to store/read directory entries is (Highlight or Circle the correct answer):
   struct entry
   struct utmpx
   struct dirent - correct

7) **Match**; which program/command goes with which description? Write the number in front of the description (10points)
   1. cp 2.mv 3.cat 4.grep 5.|(pipe) 6.ls –l 7.ls –i 8.ls –al 9.less

__8_Used to list all entries of a directory in a long format
__9_Used to view contents of a file in text format
__2_Used to rename or relocate a file
__4_Used to filter contents of a file or output stream, according to some criteria
__7_Used to list inode numbers of files in a directory

8) What are the results of the following commands, assume you start in your home directory. Put a comment after each command.
   mkdir test – makes a directory called test
   cd test – changes directory to test
   man man > text.out – appends text.out to include man (text.out is opened it will contain MAN(1))
   cp text.out text2.out – copies text.out to text2.out
   mv text2.out ../. – moves the file text2.out up a directory
   cd .. – changes directory to the parent directory of the current directory
   rmdir test – removes directory test
   pwd – prints the path of the current working direcetory

9) What is the bit mask for the following permissions on a file?
   owner: read,write,execute group: read and write others: read and execute

   Binary: 111 110 101

   Octal: 7 6 5

10) What is the purpose of the perror call and give an example of the usage?
    It displays a description of an error that corresponds to an error code stored in the system variable errno, ex. perror(argv[1]); (perror will output the system error message corresponding to errno if argv[1] contains an invalid directory path).

11) Using which header file can we access the functions for manipulating file descriptors (Circle all that apply):
    termios.h
    file.h - correct
    fcntl.h - correct
    stat.h
    lstat.h

12) What is the purpose of argc and argv? Give an example of the use below by changing the dummy function below to print the arguments.
    argc – contains the number of arguments passed to the program.

argv – is an array of argument strings passed to the new program

From my lab 5:

```
#include <stdio.h>
#include <stdlib.h> // for exit() function
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

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

   if (argc < 2)
   {
      printf("Usage: %s <directory>\n", argv[0]);
      printf("The directory might exist or not.\n");
      exit(1);
   }

   fd = chdir(argv[1]); // changes the directory, argv[1] will contain a path to the directory
   // printf will output user's formatted string, while perror will output
   // the system error msg corresponding to errno
   if (fd != 0)
   {
      printf("Changing to directory: \"%s\" failed.\n", argv[1]);
      perror(argv[1]);
   }

   else
   {
      printf("Changing to directory: \"%s\" was successful.\n", argv[1]);
   }

   return 0;
}
```

13) How many arguments are on the line below assuming my ls is an executable and what are the vector positions based on the integer count?

```
rm -f -r -v /tmp/test.txt
```

-f, -r, and -v so three arguments and vector position of 3.

14) What is the use of the dlopen and dlclose functions and what header is required to be called in order to use them.
dlopen - loads the dynamic shared object file named by the null-terminated string filename and returns an opaque “handle” for the loaded object.
dlclose – informs the system that the symbol table handle specified by handle is no longer needed by the application.
#include <dlfcn.h>

15) When using a makefile to produce the output of code what is required at the beginning of every command line?
Tab

16) What is the purpose of a phony target? Give an example of a phony target.

To avoid a conflict with a file of the same name, and to improve performance.
Ex.
clean:
rm *.o temp

17) Aside from the target in the makefile what is define directly after the tartget definition (hint: to the right of the target after the :?

The file or files that are going to be used in the makefile command.
Ex.
ls3.o: ls3.c
gcc -g -c ls3.c

ls3.c would be what this problem is asking for.

18) What are the output fields when you run the command wc against a file?
It prints newline, word, and byte counts for files.

19) What is the purpose of the target **clean** in a makefile? Why is the target clean different from the other targets you have seen? Name what that target type is.

clean removes all executed files and when ran again, it ran without any old files.

20) In the struct stat what are the variables that has the user id, group id and the time of last access?

```
struct stat {
  uid_t    st_uid; /*user ID of owner*/
  gid_t    st_gid; /*group ID of owner*/
  time_t   st_atime; /*time of last access*/
  };
```

21) Why are POSIX variables defined in the struct stat? What does POSIX guarantee?
Declares stat() functions, as well as related functions called fstat() and lstat(). sys/stat.h

22) What is the difference between the code below:
cat /etc/passwd |grep $USERNAME > myname.txt
cat /etc/passwd |grep $USERNAME >> myname.txt
The second line of code has an additional “>", where “>” will create a new file called “myname.txt” with the information obtained, while “>>” will append the information to “myname.txt” and if the file doesn’t already exist then it’ll create it.

23) In the do_ls.c code what does the rewinddir do and what header must be included to use it?
Resets the position of the directory stream dirp to the beginning of the directory.
#include <sys/types.h>
#include <dirent.h>

24) Recall using sed, what does the following command do without using the command prompt:
echo “This is a test, and only a test” | sed ‘s/test/warning/’
Sed is known as “stream editor,” and it would parses and transforms text
It would print something like:
sed: -e expression #1, char 1: unknown command: ‘ ‘

25) If there is a / slash that is to be replaced with sed what can you do to the command to accomplish this? Note that there are a couple of ways to do this.
The “\” is used to tell the computer that “/” is to be replaced and not any regex syntax

Logan Floyd 4/27/22

26) What is required at the head of a script file and what do you have to do to the file to make sure it will run?
    #!/bin/sh
    Save the script file as filename.sh
    To run: bash filename.sh

27) Which two signals cannot be caught or intercepted?
    SIGKILL and SIGSTOP

28) What is the specific library that is used for the event driven programming?
    The NCurses library

29) If there are significant delays in an event driven program what can be done to reduce the delays? Hint: has something to do with time.
    By using the sleep command

30) What do you think of Linux/Unix programming now?
    I like Linux/Unix programming and I think it is good.