ECE 282 Lab 4

Lab Assignment Due: 6 pm, Feb 7.2017

# Command practice (include the following in the report)

**Tutorial:** [**http://eecs.mines.edu/Courses/csci274/Content/Slides/08\_redirection.pdf**](http://eecs.mines.edu/Courses/csci274/Content/Slides/08_redirection.pdf)

What do the following commands do?

1. cat (without any arguments)  
   **Hint: Ctrl-C to quit  
   Echo’s terminal input**
2. cat file\_name

**Displays file contents**

1. cat < file\_name  
     
   **Hint: Create a text file to test what it does  
   Displays contents of file**
2. cat > file\_name  
   **Overwrites file with what you input into terminal**
3. cat >> file\_name

**Appends file with what is inputted into terminal cat userlist | grep justedwa > my\_user**

Write down the commands that can do the following tasks:

1. Print the list of all users to a file named “**userlist**”  
   **Hint: ‘**who**’ and redirection  
   who > userlist**
2. Print out the line that contains your username in the file “**userlist**” to the terminal  
   **Hint 1:** grep **and ‘<’  
   Hint 2: or you can:** cat **+ pipeline +** grep **cat userlist | grep justedwa**
3. Print out the line that contains your username in the file “**userlist**” to a file named “**my\_user**”  
   **Hint: Same commands, but has additional ‘>’  
   Result:** **cat userlist | grep justedwa > my\_user**
4. Append the line that contains another person’s username of your choice from the file “**userlist**” to the file “**my\_user**”  
   **Hint: Same commands, but has additional ‘>>’**

**cat userlist | grep jshebesh >> my\_user**

1. **Result:**  
   

# Reporting error

Assume that “invisible \_file” is NOT in the current directory where you execute the following code, so it will give you the error using “**perror”** function. The purpose of this lab is to help you understand **perror.**

--perror\_file.c-------------------------

#include <stdio.h>

#include <stdlib.h> // for exit() function

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

int main(int argc, char \*argv[])

{

int fd;

if ( argc < 2 ){

printf("Usage: %s <filename>\n", argv[0]);

printf("The file might exist or not.\n");

exit(1);

}

fd = open(argv[1], O\_RDONLY);

// printf will output user's formatted string, while perror will output

// the system error msg corresponding to errno

if (fd == -1){

printf("Opening file \"%s\" failed.\n", argv[1]);

perror("WHY CAN'T I OPEN THIS FILE X( ");

}

else{

printf("The file \"%s\" was opened w/o problem.\n", argv[1]);

perror("SUCCESSFULLY OPENED THE FILE. :) ");

}

return 0;

}

Modify the code so that it will try to change to a directory rather than a access a file. What error did you get?

Submit your code to Canvas by filename **perror\_dir.c**.

**Hint: chdir() function. You may need to add additional header files. Lookup man pages for reference.**

NOTE:

From now on, the following practices will be graded as well:

1. Commenting; Your code must be commented.
2. Naming; The names that you chose for variables, constants, functions, etc must be self-descriptive.
3. Error reporting; Extensive use of perror/errno for reporting errors while doing system calls. (Think in advance)

# Again, ls

In the previous lab, you examined the ls2.c program. It is a version of ls -lR (it is oneR). Now, revise the program so it prints out listing of all subdirectories and contents. Notice the indentation for each directory entry. For each level deeper, indent the entry with two spaces before printing out.

**Hint: Use the method found in the source codes of the book**

Submit the code to Canvas. Separate the source code in files and use makefile for compilation. And write the functions and lines that you added in the lab report.

**Result:**

.

..

total 19

15748396 drwxr-xr-x. 3 hessamla student 5 Feb 9 12:30 dhl

15748395 drwxr-xr-x. 4 hessamla student 4 Feb 9 12:30 dir1

15748393 -rwxr-xr-x. 1 hessamla student 6963 Feb 9 12:27 perror

15748394 -rw-r--r--. 1 hessamla student 359 Feb 9 12:27 perror\_sample.c

./dhl:

total 3

15748399 drwxr-xr-x. 2 hessamla student 4 Feb 9 12:30 abc

15748397 -rw-r--r--. 1 hessamla student 0 Feb 9 12:30 x

15748398 -rw-r--r--. 1 hessamla student 0 Feb 9 12:30 y

./dhl/abc:

total 1

15748400 -rw-r--r--. 1 hessamla student 0 Feb 9 12:30 ax

15748401 -rw-r--r--. 1 hessamla student 0 Feb 9 12:30 thisFile

./dir1:

total 3

15748403 drwxr-xr-x. 2 hessamla student 4 Feb 9 12:31 k

15748402 drwxr-xr-x. 2 hessamla student 3 Feb 9 12:31 m

./dir1/k:

total 1

15748404 -rw-r--r--. 1 hessamla student 0 Feb 9 12:31 newfile

15748405 -rw-r--r--. 1 hessamla student 0 Feb 9 12:31 round

./dir1/m:

total 1

15748406 -rw-r--r--. 1 hessamla student 0 Feb 9 12:31 mfile