SSN College of Engineering

Department of Computer Science and Engineering

UCS1411 – Operating Systems Laboratory

II Year CSE - A Section (IV Semester)

Academic Year 2019-20

Exercise – 2- Simulation of system commands using system calls

Objective:

1. To develop a C program to implement the cp, ls, grep commands (with some options) using system calls.

Sample Learning Outcome:

1. Implement the various system commands like cp, grep, ls, head, tail, wc using system calls

2. Learn to process command line arguments and error handling mechanism

3. Understand the relation between the system calls and commands

Best Practices:

1. Algorithm design

2. Naming convention – for file names, variables

3. Comment usage at proper places

4. Prompt messages during reading input and displaying output

5. Error handling mechanisms for failures in system calls

6. Incremental program development

7. Modularity

8. All possible test cases in output

AIM:
To develop a C program to implement the cp, ls, grep commands (with some options) using system calls.

cp command: basic cp, -i
To copy a file into another
ls command: basic ls, -1, -R
To list all files in the directory
grep command: basic grep, -c, -v, -n
To search the given pattern in the file

Procedure for cp:
1. The arguments should be obtained in command line and error messages should be printed if they are not sufficient.
2. Use open, read, write, creat ,close system calls to do the following.
mycp sourcefilename destinationfilename
-copies source file to destination file
3. The failure messages for opening a file, creating a file should be intimated.
Note: mycp is the user programs implementing cp.

Procedure for ls:
1. To view the files in a directory include dirent.h that helps for opening, reading, closing a directory.
2. Open the user named directory giving specific path using opendir system call. This returns a pointer to a DIR data structure that represents a directory.
3. Can even use "." to represent the current working directory.
4. Traverse the directory entries using readdir system call. readdir () returns a pointer to a dirent structure whose member d_name contains the name of the current file.
5. Output the entries of directory.
6. Close the directory pointer
NOTE: Use open, read, write, creat ,close, opendir, readdir, closedir system calls wherever necessary.

Procedure for grep:
1. Open the command line specified file using the required system call.
2. Read the contents iteratively till the end of the file and compare it with the pattern you are searching for.
3. If word found print the line on to the display.
4. Count the number of occurrences and display it finally.
5. Close the file descriptor.
NOTE: Use open, read, write, creat ,close system calls wherever necessary.

SAMPLE INPUT/OUTPUT:
cp:
Source.txt:
SSN COLLEGE OF ENGINEERING

target.txt:
SSN NAGAR
KALAVAKKAM

$ ./mycp source.txt target.txt
FILE COPIED!

ls:
$ ./myls lab

OUTPUT:
.
..
diros
diros.zip
Ex-3-cp-cat.doc
Ex-3-cp-cat.pdf
Ex-4-ls-grep.doc
fork.pdf
grep.doc
prgs.doc
sys-call prgs.doc

grep:

$ ./mygrep pattern filename

OUTPUT:
Display the contents of the file that has the pattern in it