**ASSIGNMENT 2**

**IMPLEMENTATION OF CP, LS AND GREP COMMANDS**

**SREYAS V**

**185001162**

**Cse C**

CP

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/types.h>

int copyfile(char \*src, char \*dest); //function to copy contents from one file to another

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

{

if (argc < 3 || argc > 4)

{

printf("Require 3 or 4 arguments only.");

}

else if (argc == 3) //default copy

{

copyfile(argv[1], argv[2]);

}

else //interactive mode copy

{

if (strcmp(argv[1], "-i") != 0)

{

printf("\nInvalid Option!");

return 0;

}

else

{

char c = 'N';

printf("\n%s's contents are going to be copied to %s.\nEnter (y/Y) to COPY, (n/N) to EXIT\n", argv[2], argv[3]);

scanf(" %c", &c);

if (c != 'Y' && c != 'y')

return 0;

else

copyfile(argv[2], argv[3]);

}

}

}

int copyfile(char \*srcfile, char \*destfile)

{

int src, dest, count;

char \*buffer[1024];

src = open(srcfile, O\_RDONLY); //open src in read only mode

if (src == -1)

{

printf("\nSource File Not Found!");

close(src);

return 0;

}

else

{

dest = open(destfile, O\_WRONLY | O\_CREAT, S\_IRWXU); //open dest in write only mode if it exists, otherwise create it, with permissions RWX for the user

while ((count = read(src, buffer, sizeof(buffer))) != 0)

write(dest, buffer, count);

printf("\nCopied Successfully!\n");

}

close(src);

close(dest);

}

LS

#include<stdio.h>

#include<stdlib.h>

#include<dirent.h>

#include<sys/types.h>

#include<string.h>

/\*void ls\_rec(char \*name)

{

char path[1000];

struct dirent \*de;

DIR \*dir=opendir(name);

if(dir==NULL)

return;

while((de=readdir(dir))!=NULL)

{

if(strcmp(de->d\_name,".")!=0 && strcmp(de->d\_name,"..")!=0)

{

printf("%s\n",de->d\_name);

strcpy(path,name);

strcat(path,"/");

strcat(path,de->d\_name);

ls\_rec(path);

}

}

closedir(dir);

}

\*/

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

{

struct dirent \*de;

if(argc==2)

{

DIR \*dir=opendir(argv[1]);

if(dir==NULL)

{

printf("Unable to open directory");

return 0;

}

while((de = readdir(dir)))

if (de->d\_name[0]!='.')

printf("%s ",de->d\_name);

printf("\n");

closedir(dir);

return 0;

}

else if(argc==3)

{

DIR \*dir=opendir(argv[2]);

if(dir==NULL)

{

printf("Unable to open directory");

return 0;

}

de=readdir(dir);

if(argv[1][1]=='1')

{

while(de)

{

if (de->d\_name[0]!='.')

printf("%s\n",de->d\_name);

de=readdir(dir);

}

}

else if(argv[1][1]=='a')

{

while(de)

{

printf("%ld %s\n",de->d\_ino,de->d\_name);

de=readdir(dir);

}

}

else if(argv[1][1]=='r')

{

char arr[100][100];

int c = -1;

while(de)

{

if(de->d\_name[0] != '.')

strcpy(arr[++c],de->d\_name);

de=readdir(dir);

}

for(;c > -1;c--)

printf("%s\n",arr[c]);

}

else

printf("Invalid argument\n");

/\*else if(argv[1][1]=='r')

{

char arr[50][50];

int count=0;

DIR \*dir=opendir(argv[2]);

if(dir==NULL)

{

printf("Unable to open directory");

return 0;

}

while((de = readdir(dir)))

{

printf("%s\n",de->d\_name);

//strcpy(arr[count],de->d\_name);

//count++;

}

for(int i=count-1;i>=0;i++)

{

printf("%s\n",arr[i]);

}

closedir(dir);

}\*/

closedir(dir);

return 0;

}

else

printf("Invalid no. of arguments\n");

}

GREP

#include <stdio.h>

#include <string.h>

#include <dirent.h>

#include <unistd.h>

#include <fcntl.h>

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

{

int source,count=0;

char line[100], buffer;

char \*pattern;

if (argc == 3)

{

pattern = argv[1];

source = open(argv[2], O\_RDONLY);

if (source == -1)

{

fprintf(stderr, "Unable to open file");

return 1;

}

while (read(source, &buffer, 1))

{

if (buffer == '\n')

{

line[count] = '\0';

if (strstr(line, pattern))

printf("%s\n", line);

count = 0;

for (int i = 0; i < 100; i++)

line[i] = 0;

}

else

line[count++] = buffer;

}

return 0;

}

if (argc == 4)

{

pattern = argv[2];

source = open(argv[3], O\_RDONLY);

if (source == -1)

{

fprintf(stderr, "Unable to open file");

return 1;

}

if (strcmp(argv[1], "-c") == 0)

{

int match = 0;

while (read(source, &buffer, 1))

{

if (buffer == '\n')

{

line[count] = '\0';

if (strstr(line, pattern))

match++;

count = 0;

for (int i = 0; i < 100; i++)

line[i] = 0;

}

else

line[count++] = buffer;

}

printf("No of matches: %d\n", match);

}

else if (strcmp(argv[1], "-n") == 0)

{

int c=1;

while (read(source, &buffer, 1))

{

if (buffer == '\n')

{

line[count] = '\0';

if (strstr(line, pattern))

printf("%d:%s\n", c, line);

count = 0;

for (int i = 0; i < 100; i++)

line[i] = 0;

c++;

}

else

line[count++] = buffer;

}

}

else if (strcmp(argv[1],"-v") == 0)

{

while (read(source, &buffer, 1))

{

if (buffer == '\n')

{

line[count] = '\0';

if (!strstr(line, pattern))

printf("%s\n",line);

count = 0;

for (int i = 0; i < 100; i++)

line[i] = 0;

}

else

line[count++] = buffer;

}

}

else

{

fprintf(stderr, "Invalid argument\n");

return 1;

}

}

else

printf("Invalid no. of arguments\n");

close(source);

}