**ExNo: 4 SIMULATION OF UNIX COMMANDS**

**Date :21.03.2021**

**Aim**

To simulate unix commands used to work with Unix environment.

**1.Write a program using ls command to list the filenames and subdirectories.**

**Algorithm**

1. Store path of current working directory using getcwd system call.

2. Scan directory of the stored path using scandir system call and sort the resultant array of structure.

3. Display dname member for all entries if it is not a hidden file.

4. Stop.

**Program**

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <dirent.h>

void main()

{

char \*curr\_dir = NULL;

DIR \*dp = NULL;

struct dirent \*dptr = NULL;

int count = 0;

curr\_dir = getenv ("PWD");

if(curr\_dir == NULL)

{

printf("\n ERROR : Could not get the working directory\n");

exit(-1);

}

dp = opendir((const char\*)curr\_dir);

if(dp == NULL)

{

printf("\n ERROR : Could not open the working directory\n");

exit(-1);

}

printf("\n");

for(count = 0; (dptr = readdir(dp)) != NULL; count++)

{

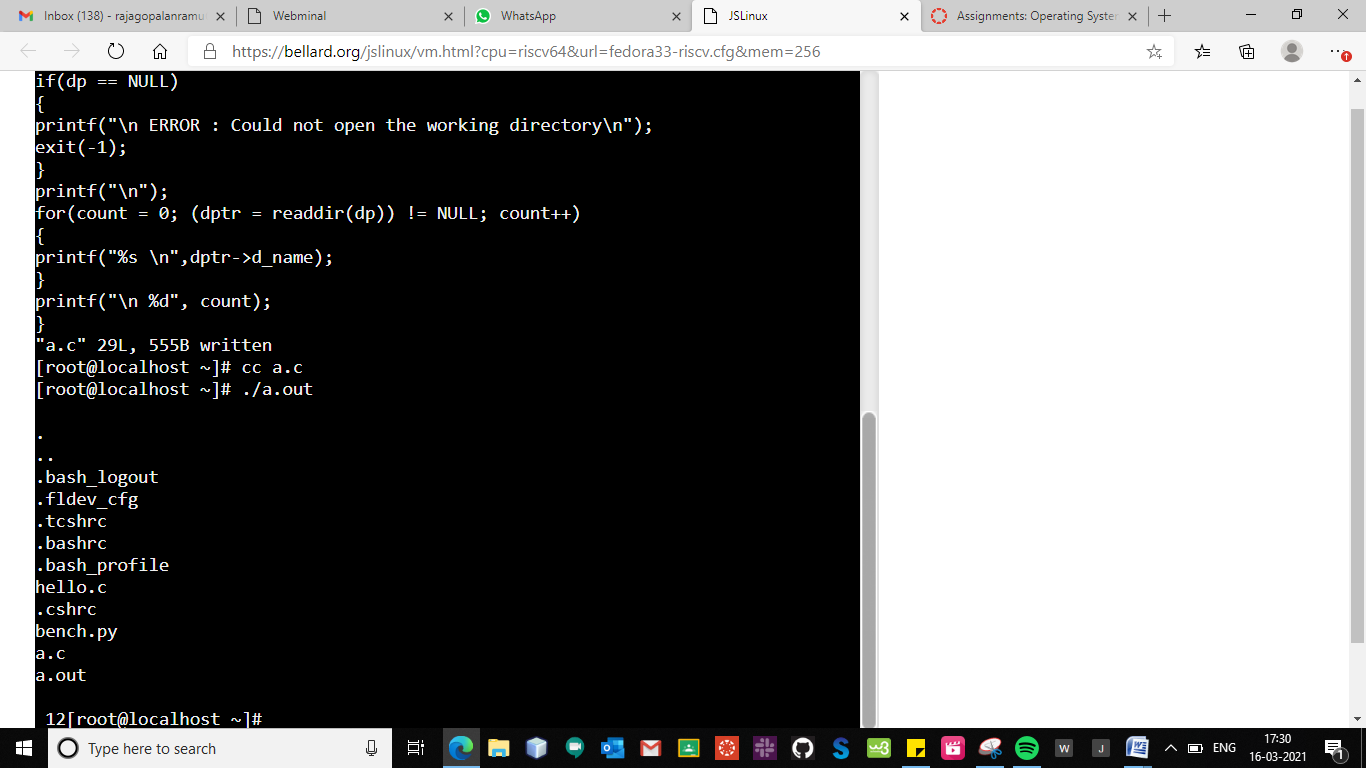
printf("%s \n",dptr->d\_name);

}

printf("\n %d", count);

}

**Output**



**2.Write a program using grep command to list the lines containing the search text.**

**Algorithm**

1. Get filename and search string as command-line argument.

2. Open the file in read-only mode using open system call.

3. If file does not exist, then stop.

4. Let length of the search string be n.

5. Read line-by-line until end-of-file

a. Check to find out the occurrence of the search string in a line by examining characters in the range 1–n, 2–n+1, etc.

b. If search string exists, then print the line.

6. Close the file using close system call.

7. Stop.

**Program**

#include<stdio.h>

#include<string.h>

void main()

{

FILE \*fptr;

char ch;

int i;

char p[10], a[50];

fptr=fopen("input.txt","w");

printf("Enter the data to be stored in the file\n");

scanf("%c",&ch);

while(ch!='$')

{

fprintf(fptr,"%c",ch);

scanf("%c",&ch);

}

fclose(fptr);

printf("Enter the pattern to be searched");

scanf("%s",p);

fptr=fopen("input.txt","r");

i=0;

while(!feof(fptr))

{

ch=getc(fptr);

if( ch!='\n')

a[i] = ch;

else

{

a[i]='\0';

if(strncmp(a,p,strlen(p))==0)

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

i=-1;

}

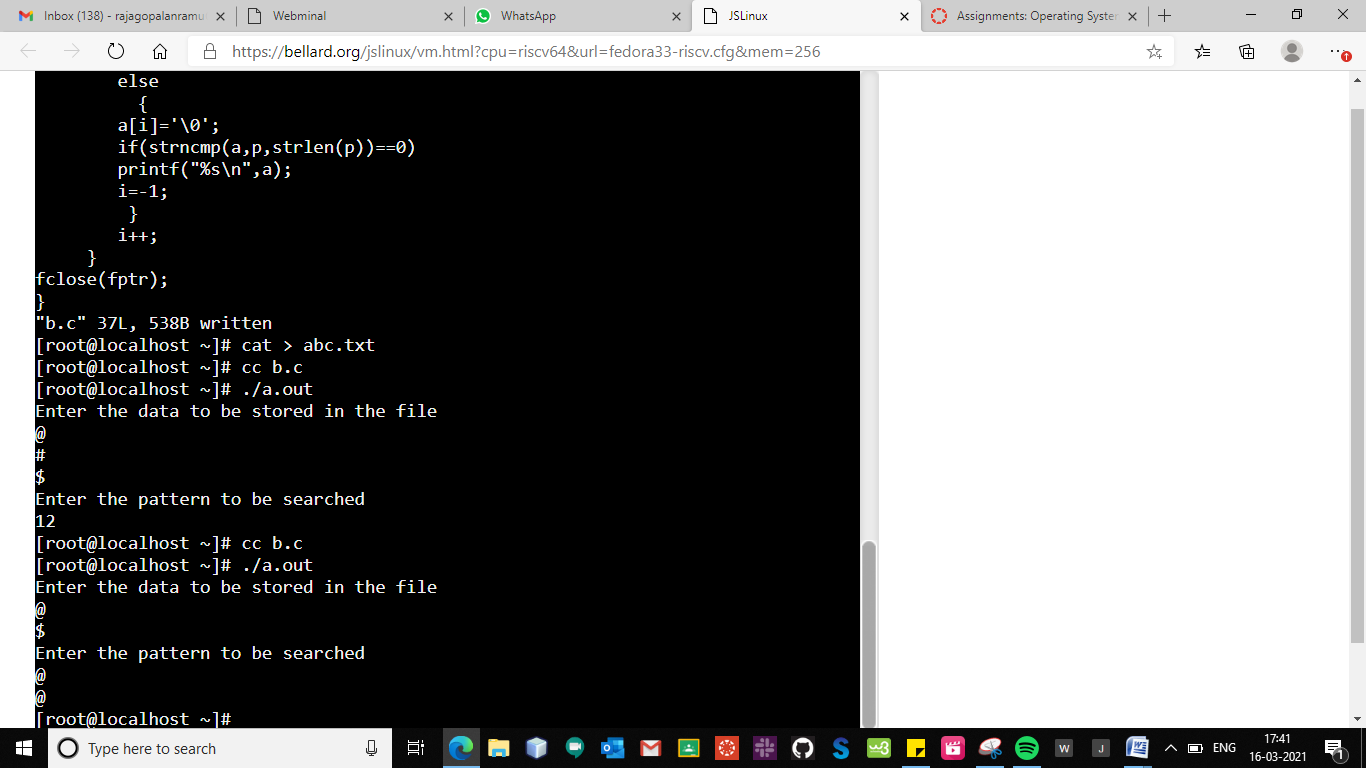
i++;

}

fclose(fptr);

}

**Output**



**3.Write a program using cp command tocopy the file using file I/O and also verify the contents of the both files.**

**Algorithm**

1. Get source and destination filename as command-line argument.

2. Declare a buffer of size 1KB

3. Open the source file in readonly mode using open system call.

4. If file does not exist, then stop.

5. Create the destination file using creat system call.

6. If file cannot be created, then stop.

7. File copy is achieved as follows:

a. Read 1KB data from source file and store onto buffer using read system call.

b. Write the buffer contents onto destination file using write system call.

c. If end-of-file then step 8 else step 7a.

8. Close source and destination file using close system call.

9. Stop.

**Program**

#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <sys/stat.h>

#define SIZE 1024

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

{

int src, dst, nread;

char buf[SIZE];

if (argc != 3)

{

printf("Usage: cc copy.c \n");

printf("Usage: ./a.out<filename><newfile> \n");

exit(-1);

}

if ((src = open(argv[1], O\_RDONLY)) == -1)

{

perror(argv[1]);

exit(-1);

}

if ((dst = creat(argv[2], 0644)) == -1)

{perror(argv[1]);

exit(-1);

}

while ((nread = read(src, buf, SIZE)) > 0)

{

if (write(dst, buf, nread) == -1)

{

printf("can't write\n");

exit(-1);

}

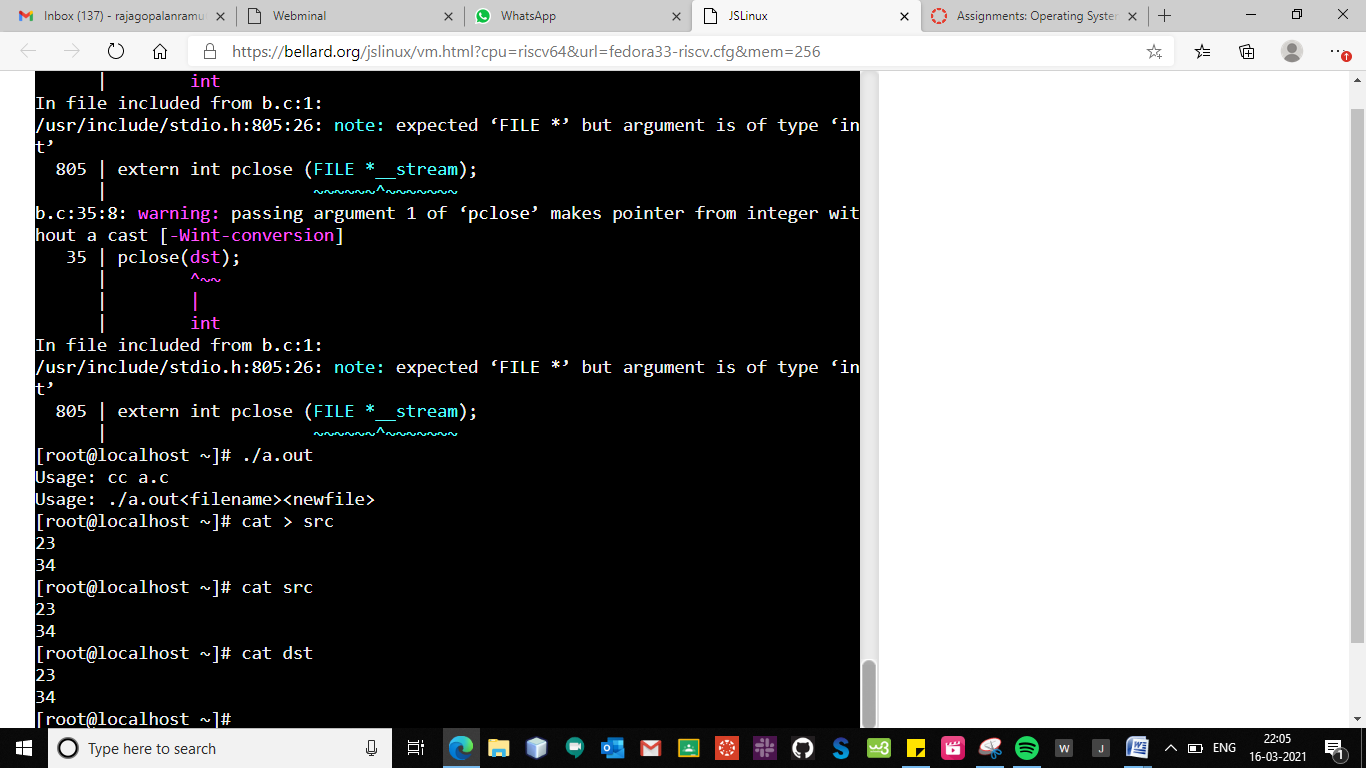
}

close(src);

close(dst);

}

**Output**



**4.Write a program using rm command to delete the file and verify it using ls command.**

**Algorithm**

1. Get filename as command-line argument.

2. Open the file in read-only mode using read system call.

3. If file does not exist, then stop.

4. Close the file using close system call.

5. Delete the file using unlink system call.

6. Stop

**Program**

#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

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

{

int fd;

if (argc != 2)

{

printf("Usage: cc del.c\n");

printf("Usage: ./a.out<filename>\n");

exit(-1);

}

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

if (fd != -1)

{

close(fd);

unlink(argv[1]);

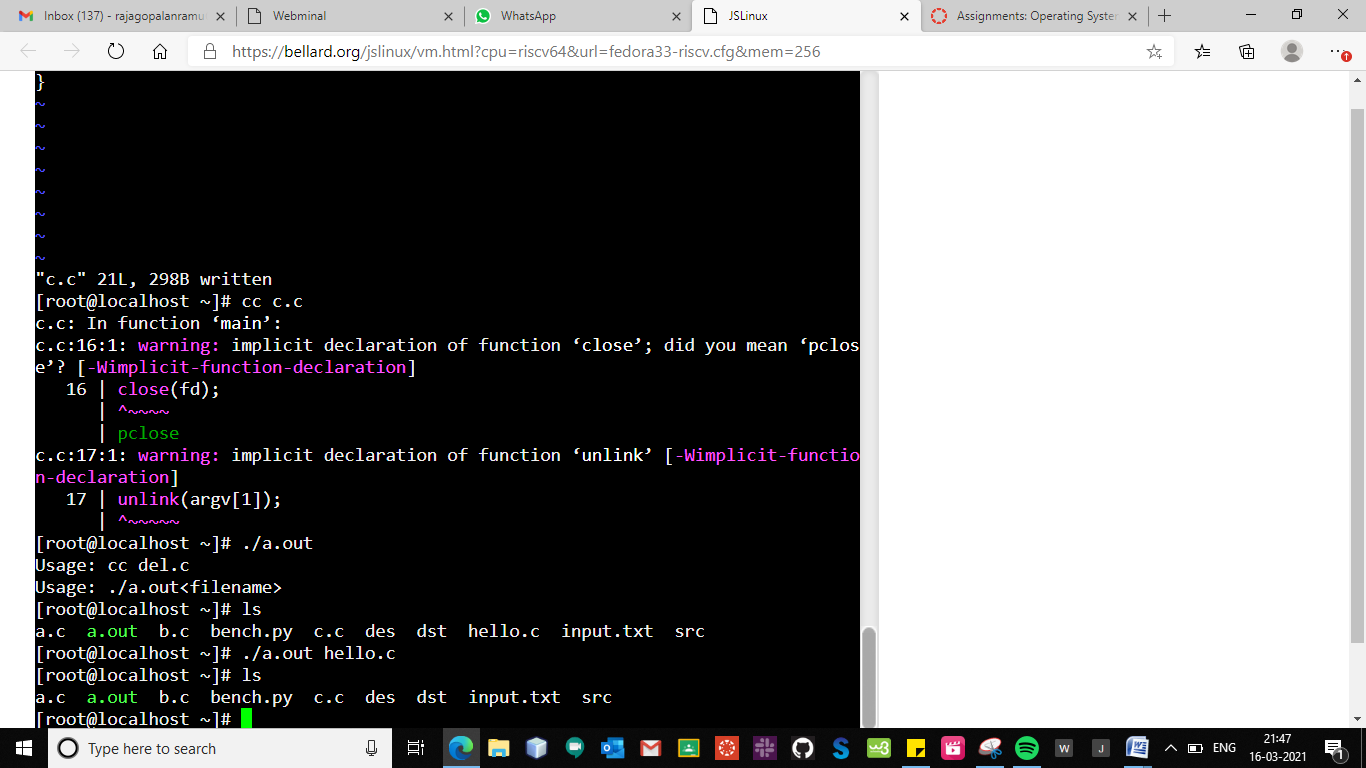
}

else

perror(argv[1]);

}

**Output**



|  |  |
| --- | --- |
| **Observation(20)** |  |
| **Record(5)** |  |
| **Total(25)** |  |
| **Initial** |  |

**Result:**

Thus the simulation of unix commands were executed and outputs were noted.