

Name: Chillara V L N S Pavana Vamsi

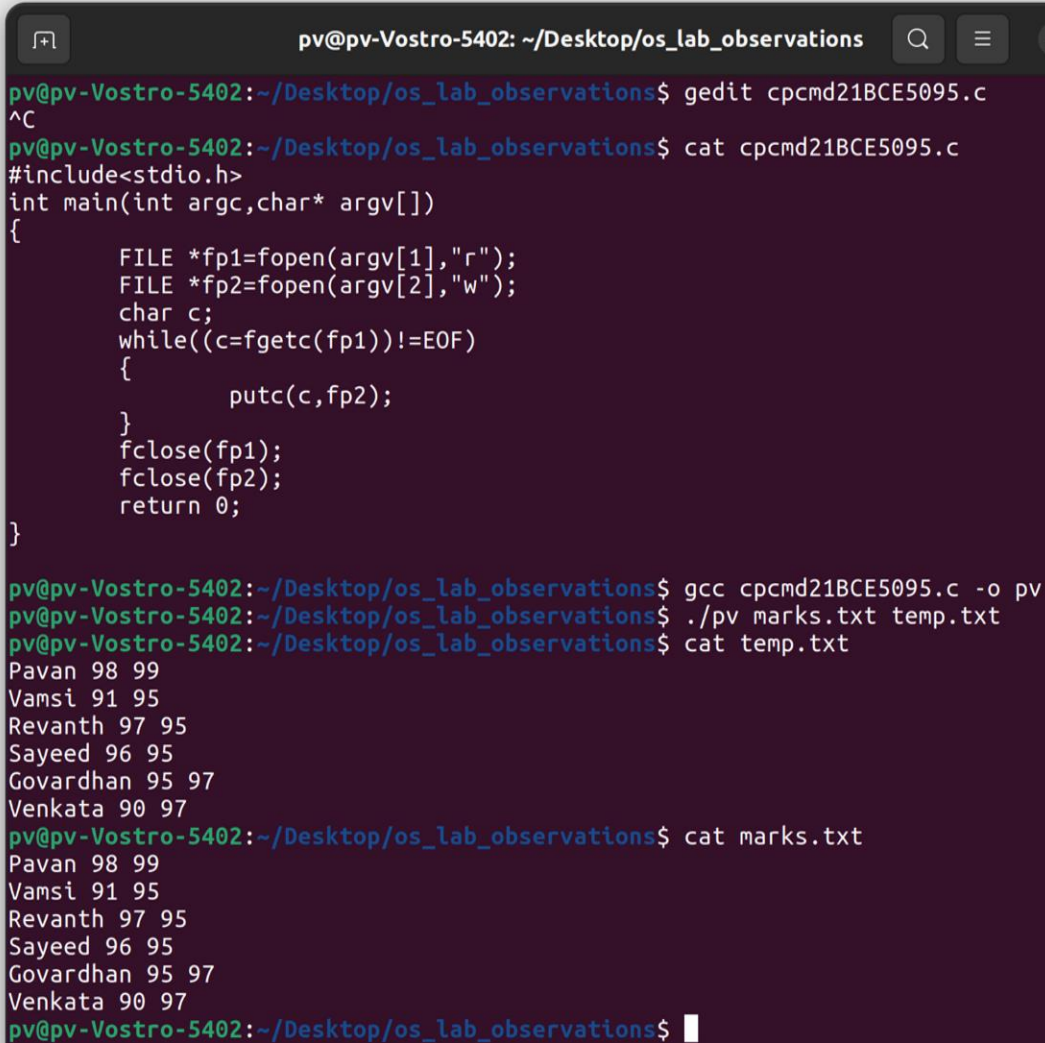
Reg.no: 21BCE5095

Date: 05/1/2023

Faculty: M Sivagami

L21_L22_Lab3 Exercises

1. Simulate CP Linux command using C



```
pv@pv-Vostro-5402: ~/Desktop/os_lab_observations
pv@pv-Vostro-5402:~/Desktop/os_lab_observations$ gedit cpcmd21BCE5095.c
^C
pv@pv-Vostro-5402:~/Desktop/os_lab_observations$ cat cpcmd21BCE5095.c
#include<stdio.h>
int main(int argc,char* argv[])
{
    FILE *fp1=fopen(argv[1],"r");
    FILE *fp2=fopen(argv[2],"w");
    char c;
    while((c=fgetc(fp1))!=EOF)
    {
        putc(c,fp2);
    }
    fclose(fp1);
    fclose(fp2);
    return 0;
}

pv@pv-Vostro-5402:~/Desktop/os_lab_observations$ gcc cpcmd21BCE5095.c -o pv
pv@pv-Vostro-5402:~/Desktop/os_lab_observations$ ./pv marks.txt temp.txt
pv@pv-Vostro-5402:~/Desktop/os_lab_observations$ cat temp.txt
Pavan 98 99
Vamsi 91 95
Revanth 97 95
Sayeed 96 95
Govardhan 95 97
Venkata 90 97
pv@pv-Vostro-5402:~/Desktop/os_lab_observations$ cat marks.txt
Pavan 98 99
Vamsi 91 95
Revanth 97 95
Sayeed 96 95
Govardhan 95 97
Venkata 90 97
pv@pv-Vostro-5402:~/Desktop/os_lab_observations$
```

2. Simulate MV Linux command using C

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit moveCmd21BCE5095.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat moveCmd21BCE5095.c
#include<stdio.h>
int main(int argc,char* argv[])
{
    FILE *fp1=fopen(argv[1],"r");
    FILE *fp2=fopen(argv[2],"w");
    char c;
    while((c=fgetc(fp1))!=EOF)
    {
        putc(c,fp2);
    }
    fclose(fp1);
    remove(argv[1]);
    fclose(fp2);
    return 0;
}

pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc moveCmd21BCE5095.c -o pv
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./pv temp.txt temp2.txt
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat temp2.txt
Pavan 98 99
Vamsi 91 95
Revanth 97 95
Sayeed 96 95
Govardhan 95 97
Venkata 90 97
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat temp.txt
cat: temp.txt: No such file or directory
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$

```

3. Perform arithmetic operations using command line arguments

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit arthCmd21BCE5095.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat arthCmd21BCE5095.c
#include<stdio.h>
#include<stdlib.h>
int main(int argc,char *argv[])
{
    int n1=atoi(argv[1]);
    int n2=atoi(argv[2]);
    int res=n1+n2;
    printf("Sum:%d+%d=%d\n",n1,n2,res);
    res=n1-n2;
    printf("Diffrence:%d-%d=%d\n",n1,n2,res);
    res=n1*n2;
    printf("Multiplication:%d*d=%d\n",n1,n2,res);
    res=n1/n2;
    printf("Division:%d/%d=%d\n",n1,n2,res);
    return 0;
}

pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc arthCmd21BCE5095.c -o arth
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./arth 18 7
Sum:18+7=25
Diffrence:18-7=11
Multiplication:18*7=126
Division:18/7=2
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$

```

4. Check whether the given string is palindrome or not and ensure to take the input while executing the program

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit palindromeString21BCE5095.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat palindromeString21BCE5095.c
#include<stdio.h>
#include<string.h>
int main(int argc,char *argv[])
{
    int flag=0;
    char *str=argv[1];
    int length=strlen(str);
    for (int i=0;i<length/2;i++)
    {
        if(str[i]!=str[length-i-1])
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    printf("Not a Palindrome String.\n");
    else
    printf("Palindrome String.\n");
    return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc palindromeString21BCE5095.c
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out malayalam
Palindrome String.
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out pavana
Not a Palindrome String.
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$

```

5. Create 3 child processes from the same parent process and show the child processes are created from the same parent process.

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gedit parentCheck21BCE5095.c
^C
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ cat parentCheck21BCE5095.c
#include<stdio.h>
#include<unistd.h>
int main()
{
    for(int i=0;i<3;i++)
    {
        int k=fork();
        if(k==0)
        {
            printf("Child %d process id: %d and his parent process id: %d\n",i+1,getpid(),getppid());
            return 0;
        }
    }
    return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ gcc parentCheck21BCE5095.c
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$ ./a.out
Child 1 process id: 15538 and his parent process id: 15537
Child 2 process id: 15539 and his parent process id: 15537
Child 3 process id: 15540 and his parent process id: 15537
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations$

```

6. Discuss the use of Command line arguments in C

- Command line arguments are parameters that are passed to a program when it is executed from the command line. This helps us to control the values from outside instead of declaring everything inside the code which is fixed and we can't modify that from outside.
- In the code we should make some change in the main function by passing argument count and argument vector as parameters to that main function
- `int main (int argc, char* argv [])`
- `argv [0]` represents program name
- `argv [1]` to etc represents the arguments you passed during the execution in an order.