Name : Omkar Arun Shinde PRN No. : 122B1B258

Assignment No. 02

Problem Statement : Write a program to simulate use of Linux commands like cp, grep with the usage of fork () and exec () system calls. Also show the usage of wait (), getpid () and exit () system calls.

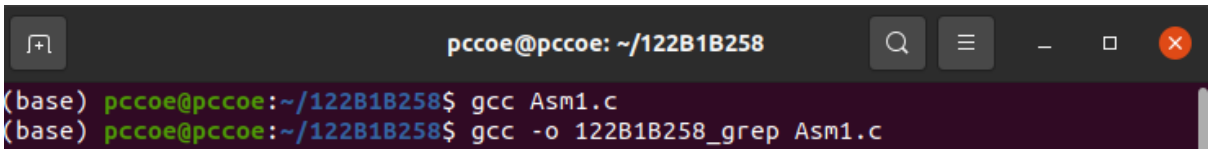
GREP Command Program

Code :

```
#include<stdio.h>

#include<string.h>


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

   if(argc != 3) {

      printf("You can not enter more or less than 3 arguments.");

      return 0;

      }

      char* fn;

      char* pat;

      char line[5000];

      FILE* fp;

      char* match;

      pat = argv[1];

      fn = argv[2];

      fp = open(fn, O_RDONLY);

      while(!feof(fp)) {

         fgets(line, 5000, fp);

         match = strstr(line, pat);

         if(match) {

            *match = '\0';
```

```
            printf("%s", line);
            printf("\033[31m%s\033[0m", pat);
            printf("%s", match + strlen(pat));
        }
    }
  close(fp);
  return 0;
}
```

Compilation :



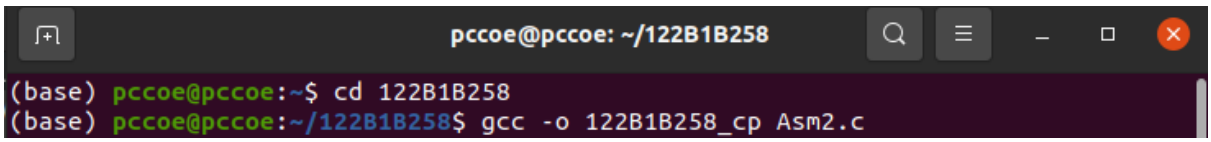
CP Command Program

Code :

```
#include<stdio.h>
#include<string.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<unistd.h>

int main(int argc, char* argv[]) {
  if(argc != 3) {
    printf("You can not enter more or less than 3 arguments.");
    return 0;
  }
```

```
    char* fn1;
    char* fn2;
    char line[50];
    int fp1;
    int fp2;
    int n;
    fn1 = argv[1];
    fn2 = argv[2];
    fp1 = open(fn1, O_RDONLY);
    fp2 = open(fn2, O_WRONLY);
    while ((n = read(fp1, line, 50)) > 0) {
        write(fp2, line, n);
    }
    close(fp2);
    close(fp1);
    return 0;
}
```

Compilation :



Assignment Program

Code :

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/wait.h>
```

```
#include<unistd.h>
#include<stdlib.h>

int main() {
   int pid;
   int ppid;
   int fork_var;
   int ch;

   pid = getpid();
   ppid = getppid();
   printf("You are in a Parent Process\n");
   printf("The PID of this process is %d\n", pid);
   printf("The PID of parent process is %d\n", ppid);

   do {
      printf("\nYou are welcome!!!\nChoose any one option:\n1. GREP Command\n2. CP
Command\n3. Exit\n");
      printf("Enter your choice: ");
      scanf("%d", &ch);

      switch(ch) {
      case 1: {
         fork_var = fork();
         if (fork_var == 0) {
            printf("You are in a child process.\n");
            pid = getpid();
            ppid = getppid();
```

```
        printf("The PID of child process is %d\n", pid);
        printf("The PID of parent process is %d\n", ppid);
        char pat[10];
        char fl[10];
        printf("Enter a file name: ");
        scanf("%s", fl);
        printf("Enter a pattern: ");
        scanf("%s", pat);
        printf("Running GREP command:\n");
        execlp("./122B1B258_grep", "122B1B258_grep", pat, fl, NULL);
        exit(1);
    } else if (fork_var > 0) {
        wait(NULL);
    } else {
        perror("Fork failed");
    }
    break;
}
case 2: {
    fork_var = fork();
    if (fork_var == 0) {
        printf("You are in a child process.\n");
        pid = getpid();
        ppid = getppid();
        printf("The PID of child process is %d\n", pid);
        printf("The PID of parent process is %d\n", ppid);
        char fl1[10];
        char fl2[10];
```

```
            printf("Enter a source file name: ");
            scanf("%s", fl1);
            printf("Enter a destination file name: ");
            scanf("%s", fl2);
            printf("Running CP command:\n");
            execlp("./122B1B258_cp", "122B1B258_cp", fl1, fl2, NULL);
            exit(1);
        } else if (fork_var > 0) {
            wait(NULL);
        } else {
            perror("Fork failed");
        }
        break;
    }
    case 3: {
        printf("Thank you!!!\n");
        exit(1);
    }
    default: {
        printf("You entered wrong option!!! Please try again.\n");
        break;
    }
    }
  } while(ch != 3);

  return 0;
}
```

Output :

```
pccoe@pccoe: ~/122B1B258
(base) pccoe@pccoe:~/122B1B258$ gcc -o 122B1B258_OSL1 OSL1.c
(base) pccoe@pccoe:~/122B1B258$ ./122B1B258_OSL1
You are in a Parent Process
The PID of this process is 4448
The PID of parent process is 4215

You are welcome!!!
Choose any one option:
1. GREP Command
2. CP Command
3. Exit
Enter your choice: 1
You are in a child process.
The PID of child process is 4449
The PID of parent process is 4448
Enter a file name: test.txt
Enter a pattern: opqrs
Running GREP command:
My SGPA of 5th Sem is 9.53 and CGPA is 9.23. So My pattern for 1st assignment is
abcdefghijklmnopqrstuvwxyz. So lets check whether our program is working or not.
```

```
pccoe@pccoe: ~/122B1B258
abcdefghijklmnopqrstuvwxyz. So lets check whether our program is working or not.

You are welcome!!!
Choose any one option:
1. GREP Command
2. CP Command
3. Exit
Enter your choice: 2
You are in a child process.
The PID of child process is 4471
The PID of parent process is 4448
Enter a source file name: test.txt
Enter a destination file name: copy.txt
Running CP command:

You are welcome!!!
Choose any one option:
1. GREP Command
2. CP Command
3. Exit
Enter your choice: 3
Thank you!!!
(base) pccoe@pccoe:~/122B1B258$
```