Hasib Ar Rafiul Fahim

2019-1-60-036

Lab Tasks

Lab 3:

Task 6:

#include <stdio.h>

#include <sys/types.h>

#define MAX\_COUNT 200

void ChildProcess(void);

void ParentProcess(void);

void main(void)

{

pid\_t pid;

pid pPid,cPid;

pid = fork();

if (pid == 0)

ChildProcess();

cPid=getpid();

else

ParentProcess();

pPid=getpid();

}

void ChildProcess(void)

{

int i;

wait(5);

for (i = 1; i <= MAX\_COUNT; i++){

printf("Hello Parent\n");

printf("%d",pPid);

}

}

void ParentProcess(void)

{

int i;

for (i = 1; i <= MAX\_COUNT; i++){

printf("Hello Child \n");

printf("%d",cPid);

}

}

Lab 5:

lab03\_prog01.c :

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

    if(argc!=2){

        printf(“Invalid number of arguments.\n”);

        printf(“Usage: PROG\_NAME FILE\_NAME\n”);

        exit(1);

    }

    char \*filename = argv[1];

    printf(“Creating a file %s ... \n”; **argv**);

    creat(**argv**,0644);

// open file for writing only

    int fd = open(**argc**,**argv**,0);

    int sz; //actual size of the file

    if(fd==-1){

        printf(“Error opening file\n”);

**close(fd);** // exit from program

    }

    printf(“File Descriptior: %d\n”, fd);

    char str[1024];

strcpy(str,”Hello World”);

**argv** = write(fd, **str**, 1024);

    close(fd);

    char buf[1024]; //hold value to be read

    fd = open(filename,O\_RDONLY,0);

// include error checking here

if(fd == **0**){

printf(“**End of the line**”);

**close(fd)**; // exit from program

    }

    read(**fd**, **buf**, 1024);

    printf(“%s”, **buf**);

**close(fd)**; // close file

    return 0;

}

lab03\_prog02.c:

int main(){

DIR \*dirptr=opendir(“/home/rezwan/lab03”);

    if(dirptr==NULL){

        printf(“Cannot open directory\n”);

        exit(1);

    }

    struct **DIR** \*entry = **opendir(“/home/rezwan/lab03”);**

    while(entry != NULL){

        printf(“[%ld] [%u] [%u] [%s]\n”,entry->d\_ino, entry->d\_reclen,

entry->d\_type, entry->d\_name);

        entry = **opendir(“/home/rezwan/lab03”);**

    }

    closedir(dirptr);

    return 0;

}