***CSE321 Assignment 3***

Name: Mehnaz Ara Fazal

ID: 20301295

Section: 01

Email: [mehnaz.ara.fazal@g.bracu.ac.bd](mailto:mehnaz.ara.fazal@g.bracu.ac.bd)

**C Programming**

*Task-1*

#include<stdio.h>

struct Ordered\_Item {

int quantity;

float unit\_price;

};

int main(){

struct Ordered\_Item Paratha;

printf("Quantity Of Paratha: ");

scanf("%d",&Paratha.quantity);

printf("\n");

printf("Unit Price: ");

scanf("%f",&Paratha.unit\_price);

printf("\n");

struct Ordered\_Item Vegetable;

printf("Quantity Of Vegetable: ");

scanf("%d",&Vegetable.quantity);

printf("\n");

printf("Unit Price: ");

scanf("%f",&Vegetable.unit\_price);

printf("\n");

struct Ordered\_Item Mineral\_Water;

printf("Quantity Of Mineral Water: ");

scanf("%d",&Mineral\_Water.quantity);

printf("\n");

printf("Unit Price: ");

scanf("%f",&Mineral\_Water.unit\_price);

printf("\n");

int number\_of\_people;

printf("Number of People: ");

scanf("%d",&number\_of\_people);

printf("\n");

float total\_price=0.0;

total\_price += (Paratha.quantity\*Paratha.unit\_price);

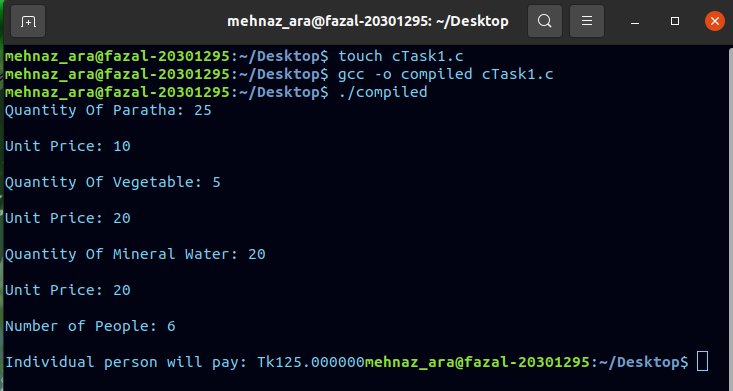
total\_price += (Vegetable.quantity\*Vegetable.unit\_price);

total\_price += (Mineral\_Water.quantity\*Mineral\_Water.unit\_price);

printf("Individual person will pay: Tk%f",total\_price/number\_of\_people);

return 0;

}



*Task-2*

#include <stdio.h>

int perfectNumber(int a);

int main(){

int initial\_number;

int last\_number;

int i;

printf("Enter starting number: ");

scanf("%d",&initial\_number);

printf("\n");

printf("Enter ending number: ");

scanf("%d",&last\_number);

printf("\n");

for(i=initial\_number;i<(last\_number+1);i++){

perfectNumber(i);

}

return 0;

}

int perfectNumber(int a){

int i;

int sum\_of\_divisors = 0;

for(i=1;i<a;i++){

if(a % i == 0){

sum\_of\_divisors += i;

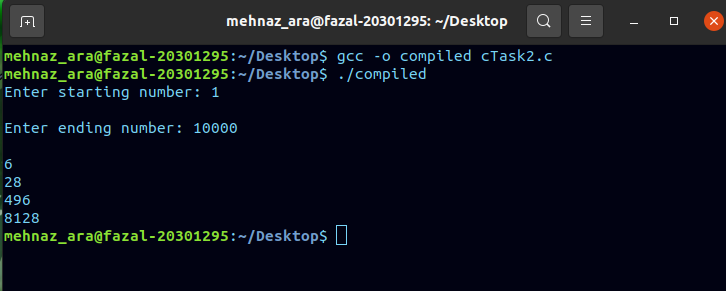
}

}

if (sum\_of\_divisors == a){

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

}

}  


**System Calls**

*Task-1*

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

FILE \*fptr;

fptr = fopen(argv[1],"w+");

char word[10];

printf("Provide a string:\n");

scanf("%s",word);

int strcompare;

strcompare = strcmp(word,"-1");

while (strcompare != 0){

fputs(word,fptr);

printf("Provide another string:\n");

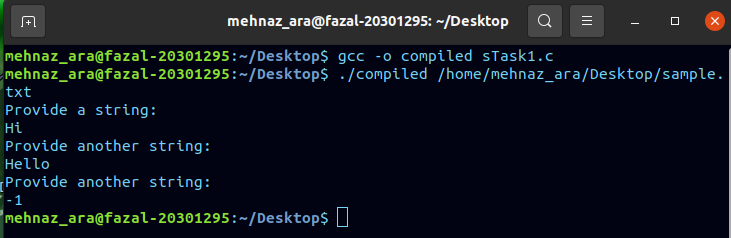
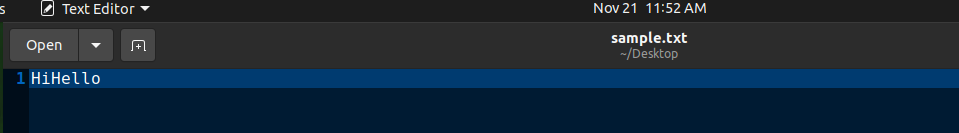
scanf("%s",word);

strcompare = strcmp(word,"-1");

}

fclose(fptr);

return 0;

}  
  


*Task-2*

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <sys/wait.h>

int main(){

pid\_t pid;

pid = fork();

if(pid == 0){

pid\_t pid2;

pid2 = fork();

if(pid2 == 0){

printf("I am the grandchild\n");

}

else if (pid2 > 0){

wait(NULL);

printf("I am the child\n");

}

}

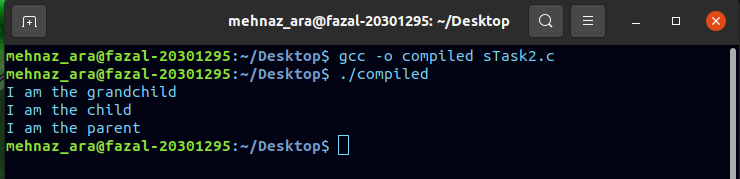
else if (pid > 1){

wait(NULL);

printf("I am the parent\n");

}

return 0;

}  


*Task-3*

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <sys/wait.h>

#include <math.h>

int noOfProcess = 0;

int main(){

int tempval;

int n = 0;

int temp = 0;

int a1,b1,c1;

pid\_t a,b,c,d,e,f;

a = fork();

b = fork();

c = fork();

if (a>0 && b>0 && c>0){

wait(NULL);

noOfProcess += 8;

if ((a%2)==1){

printf("Odd Process ID of 'a' process: %d\n",a);

a1 = 1;

n += 1;

}

else if ((a%2)==0){

printf("Even process ID of 'a' process: %d so it is ignored\n",a);

}

if ((b%2)==1){

printf("Odd Process ID of 'b' process: %d\n",b);

b1 = 1;

n += 1;

}

else if ((b%2)==0){

printf("Even process ID of 'b' process: %d so it is ignored\n",b);

}

if ((c%2)==1){

printf("Odd Process ID of c process: %d\n",c);

c1 = 1;

n += 1;

}

else if ((c%2)==0){

printf("Even process ID of 'c' process: %d so it is ignored\n",c);

}

if (a1 == 1){

d = fork();

if (d == 0){

temp = -1;

}

}

if (b1 == 1){

e = fork();

if (e == 0){

temp = -1;

}

}

if (c1 == 1){

f = fork();

if (f == 0){

temp = -1;

}

}

if (temp == 0){

wait(NULL);

tempval = pow(2,n);

noOfProcess += tempval;

printf("Total number of process is: %d\n",noOfProcess);

}

}

return 0;

}



*Task-4*

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <sys/wait.h>

int main(){

pid\_t pid;

pid = fork();

if (pid == 0){

execl("/home/mehnaz\_ara/Desktop/sort", "sort","67","48","24","33","92",NULL);

}

else if(pid>0){

wait(NULL);

printf("START");

execl("/home/mehnaz\_ara/Desktop/oddeven","oddeven","67","48","24","33","92",NULL);

}

return 0;

}

***Sort.c***

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

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

**int i,j,k,n;**

**int length = argc-1;**

**int actual\_arr[length];**

**for(i = 1;i<argc;i++){**

**actual\_arr[i-1] = atoi(argv[i]);**

**}**

**int temp\_arr[length];**

**for(j = 0;j<length;j++){**

**int max = 0;**

**int index = 0;**

**for(k = 0;k<length;k++){**

**if (actual\_arr[k]>max){**

**max = actual\_arr[k];**

**index = k;**

**}**

**}**

**actual\_arr[index] = 0;**

**temp\_arr[j] = max;**

**}**

**for(n = 0;n<length;n++){**

**actual\_arr[n] = temp\_arr[n];**

**printf("%d ",actual\_arr[n]);**

**}**

**printf("\n");**

**return 0;**

**}**

***Oddeven.c***

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

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

int i,j;

int length = argc-1;

int actual\_arr[length];

for(i = 1;i<argc;i++){

actual\_arr[i-1] = atoi(argv[i]);

}

for(j = 0;j<length;j++){

if ((actual\_arr[j]%2)==0){

printf("The number %d is even\n",actual\_arr[j]);

}

else if ((actual\_arr[j]%2)==1){

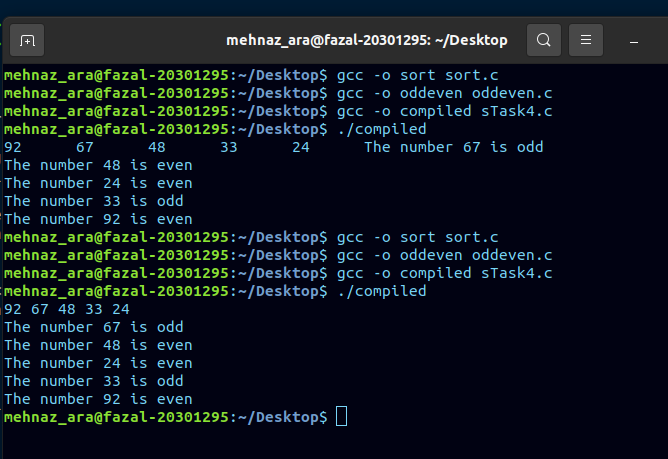
printf("The number %d is odd\n",actual\_arr[j]);

}

}

return 0;

}



*Task-5*

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <sys/wait.h>

int main(){

pid\_t pid;

int parent\_id = 0;

pid = fork();

if (pid > 0){

printf("Parent process ID : %d\n",parent\_id);

printf("Child process ID : %d\n",pid);

wait(NULL);

}

if (pid == 0){

pid\_t gc1,gc2,gc3;

gc1 = fork();

if (gc1 > 0){

gc2 = fork();

if (gc2 > 0 && gc1 > 0){

gc3 = fork();

if (gc3 > 0 && gc2 > 0 && gc1 > 0){

printf("Grand Child1 process ID : %d\n",gc1);

printf("Grand Child2 process ID : %d\n",gc2);

printf("Grand Child3 process ID : %d\n",gc3);

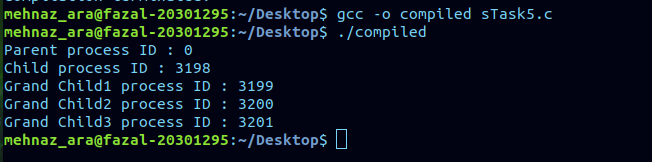
}

}

}

}

return 0;

}  


**Thread Programming**

*Task-1*

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <pthread.h>

void \*funcThread(void \*arg);

int main(){

pthread\_t t1,t2,t3,t4,t5;

int num1 = 1;

int num2 = 2;

int num3 = 3;

int num4 = 4;

int num5 = 5;

pthread\_create(&t1,NULL,funcThread, &num1);

pthread\_create(&t2,NULL,funcThread, &num2);

pthread\_create(&t3,NULL,funcThread, &num3);

pthread\_create(&t4,NULL,funcThread, &num4);

pthread\_create(&t5,NULL,funcThread, &num5);

pthread\_join(t1,NULL);

pthread\_join(t2,NULL);

pthread\_join(t3,NULL);

pthread\_join(t4,NULL);

pthread\_join(t5,NULL);

return 0;

}

void \*funcThread(void \*arg){

int number = \*(int\*)arg;

if (number==1){

printf("thread-1 running\n");

printf("thread-1 closed\n");

}

else if (number==2){

sleep(2);

printf("thread-2 running\n");

printf("thread-2 closed\n");

}

else if (number==3){

sleep(4);

printf("thread-3 running\n");

printf("thread-3 closed\n");

}

else if (number==4){

sleep(6);

printf("thread-4 running\n");

printf("thread-4 closed\n");

}

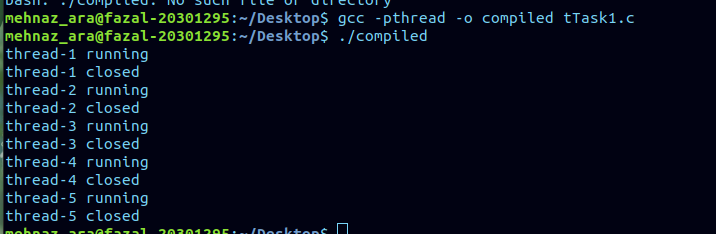
else if (number==5){

sleep(8);

printf("thread-5 running\n");

printf("thread-5 closed\n");

}

}  


*Task-2*

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <pthread.h>

int numeral = 0;

void \*funcThread(void \*arg);

int main(){

pthread\_t t1,t2,t3,t4,t5;

int num1 = 1;

int num2 = 2;

int num3 = 3;

int num4 = 4;

int num5 = 5;

pthread\_create(&t1,NULL,funcThread, &num1);

pthread\_create(&t2,NULL,funcThread, &num2);

pthread\_create(&t3,NULL,funcThread, &num3);

pthread\_create(&t4,NULL,funcThread, &num4);

pthread\_create(&t5,NULL,funcThread, &num5);

pthread\_join(t1,NULL);

pthread\_join(t2,NULL);

pthread\_join(t3,NULL);

pthread\_join(t4,NULL);

pthread\_join(t5,NULL);

return 0;

}

void \*funcThread(void \*arg){

int number = \*(int\*)arg;

if (number==1){

for (int i =0;i<5;i++){

numeral += 1;

printf("Thread 1 prints %d\n",numeral);

}

}

else if (number==2){

sleep(5);

for (int i =0;i<5;i++){

numeral += 1;

printf("Thread 2 prints %d\n",numeral);

}

}

else if (number==3){

sleep(10);

for (int i =0;i<5;i++){

numeral += 1;

printf("Thread 3 prints %d\n",numeral);

}

}

else if (number==4){

sleep(15);

for (int i =0;i<5;i++){

numeral += 1;

printf("Thread 4 prints %d\n",numeral);

}

}

else if (number==5){

sleep(20);

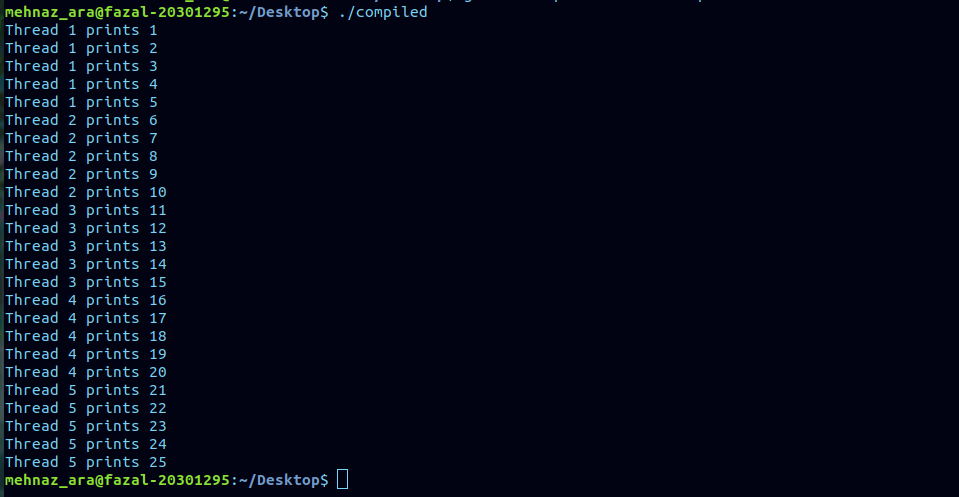
for (int i =0;i<5;i++){

numeral += 1;

printf("Thread 5 prints %d\n",numeral);

}

}

}  


*Task-3*

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <pthread.h>

int nameASCII ();

int name1,name2,name3;

void \*funcThread(void \*arg);

int main(){

pthread\_t t1,t2,t3,t4;

int num1 = 1;

int num2 = 2;

int num3 = 3;

int num4 = 4;

pthread\_create(&t1,NULL,funcThread, &num1);

pthread\_create(&t2,NULL,funcThread, &num2);

pthread\_create(&t3,NULL,funcThread, &num3);

pthread\_create(&t4,NULL,funcThread, &num4);

pthread\_join(t1,NULL);

pthread\_join(t2,NULL);

pthread\_join(t3,NULL);

pthread\_join(t4,NULL);

return 0;

}

void \*funcThread(void \*arg){

int number = \*(int\*)arg;

if (number==1){

name1 = nameASCII();

}

else if (number==2){

sleep(5);

name2 = nameASCII();

}

else if (number==3){

sleep(10);

name3 = nameASCII();

}

else if (number==4){

sleep(15);

if (name1 == name2 && name2 == name3){

printf("Eureka!");

}

else if (name1==name2||name2==name3||name1==name3){

printf("Miracle");

}

else {

printf("Hasta la vista");

}

}

}

int nameASCII (){

char name[10];

printf("Give a name: ");

scanf("%s", name);

printf("\n");

int sum = 0;

int i = 0;

while (name[i]!='\0'){

sum += name[i];

i ++;

}

return sum;

}  
