1.Develop an application for getting a name in parent and convert it into uppercase in child using shared memory.

#include<sys/ipc.h>

#define NULL 0

#include<sys/shm.h>

#include<sys/types.h>

#include<unistd.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/wait.h>

#include<ctype.h>

//#include<stdio\_ext.h>

// parent writing a char in shared memory and child reads it and prints it.

int main()

{

int pid;

char \*a,\*b,c;

int id,n,i;

char name[20];

// you can create a shared memory between parent an d child here or you can create inside them separately.

id=shmget(111,50,IPC\_CREAT | 00666);

pid=fork();

if(pid>0)

{

// id=shmget(111,50,IPC\_CREAT | 00666);

a=shmat(id,NULL,0);

printf("\nParent process is running ");

printf("\nEnter the name :");

scanf("%[^\n]s",name);

strcpy(a,name);

wait(NULL);

shmdt(a);

exit(0);

}

else

{

sleep(5);

printf("\nChild process is running "); //id=shmget(111,50,0);

b=shmat(id,NULL,0);

int n=strlen(b);

int i;

printf("\nChild converting the string to upper case :");

for(i=0;i<=n;i++)

{

printf("%c",toupper(b[i]));

}

printf("\n");

shmdt(b);

exit(0);

}

}

1. Develop an client / server application for file transfer using shared memory.

//Client program

#include<sys/ipc.h>

#define NULL 0

#include<sys/shm.h>

#include<sys/types.h>

#include<unistd.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/wait.h>

int main()

{

char \*a;

int \*b;

int id,flag;

char name[20];

int fd;

id=shmget(111,100,IPC\_CREAT | 00666);

flag=shmget(150,10,IPC\_CREAT | 00666);

a=shmat(id,NULL,0);

b=shmat(flag,NULL,0);

b[1]=0;

// if b[1]=0 means you should type the file name

// if b[1]=1 means the server should read the file name and get the contents of the file

// if b[0]=1 means the client should continue the process and displays the content of the file

// if b[0]=0 means the server should wait for the next input from the user

while(1)

{

if(b[1]==0)

{

printf("\nEnter the file name:");

gets(name)

strcpy(a,name);

b[0]=0;

b[1]=1;

}

if(b[0]==1)

{

printf("\nContents of the file: %s\n",a);

printf("\nEnter the new file name :");

gets(name);

fd=open(name, O\_WRONLY);//creates a new file

write(fd,a,strlen(a)+1);//writes the buffer onto the file

close(fd);

b[1]=0;

}

}

}

//Server program…

#include<sys/ipc.h>

#define NULL 0

#include<sys/shm.h>

#include<sys/types.h>

#include<unistd.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/wait.h>

int main()

{

char \*a;

int \*b;

int id,flag;

int fd;

char name[20],b[100];

id=shmget(111,50,IPC\_CREAT | 00666);

flag=shmget(150,10,IPC\_CREAT | 00666);

a=shmat(id,NULL,0);

b=shmat(flag,NULL,0);

while(1)

{

if(b[1]==1 && b[0]==0)

{

fd=open(a, O\_RDONLY);//opens the file

read(fd,b,100);

close(fd);

strcpy(a,b);//copies the contents to the shared memory

b[0]=1;

}

}

}

1. Develop an client/server chat application using shared memory.

//User 1 program..

#include<sys/ipc.h>

#define NULL 0

#include<sys/shm.h>

#include<sys/types.h>

#include<unistd.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/wait.h>

int main()

{

char \*a;

int \*b;

int id,flag;

char name[20];

id=shmget(111,50,IPC\_CREAT | 00666);

flag=shmget(150,10,IPC\_CREAT | 00666);

a=shmat(id,NULL,0);

b=shmat(flag,NULL,0);

b[1]=0;

// if b[1]=0 means you should write or b[1]=1 means friend should write

// if b[0]=0 means friend should print the value and b =1 means you should print the process

while(1)

{

if(b[1]==0)

{

printf("\nYOU:");

gets(name)

strcpy(a,name);

b[0]=0;

b[1]=1;

}

if(b[0]==1)

{

printf("\nFriend's msg: %s\n",a);

}

}

}

//User 2 program…

#include<sys/ipc.h>

#define NULL 0

#include<sys/shm.h>

#include<sys/types.h>

#include<unistd.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/wait.h>

int main()

{

char \*a;

int \*b;

int id,flag;

char name[20];

id=shmget(111,50,IPC\_CREAT | 00666);

flag=shmget(150,10,IPC\_CREAT | 00666);

a=shmat(id,NULL,0);

b=shmat(flag,NULL,0);

while(1)

{

if(b[0]==0)

{

printf("\nFriend's msg: %s\n",a);

}

if(b[1]==1)

{

printf("\nYou:");

gets(name);

strcpy(a,name);

b[0]=1;

b[1]=0;

}

}

}