1.Write a program to demonstrate various process related concept

#include <stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<sys/types.h>

#include<sys/wait.h>

int main()

{

int pid=fork();

if(pid==0)

{

printf("child=>PPID%d PID %d\n",getppid(),getpid());

exit(EXIT\_SUCCESS);

}

else if(pid>0)

{

printf("parent=>PID %d\n",getpid());

printf("waiting for child process\n");

wait(NULL);

printf("child proces finished");

}

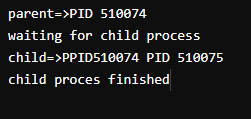
else{

printf("unable to create child process");

}

return EXIT\_SUCCESS;

}



2.write a c program to demonstrate thread related concepts

#include<stdio.h>

#include<pthread.h>

void\*routine()

{

printf("test for heading\n");

printf("test for heading\n");

printf("ending\n");

}

int main()

{

pthread\_t t1,t2;

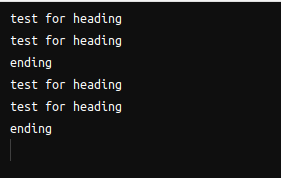
pthread\_create(&t1,NULL,&routine,NULL);

pthread\_create(&t2,NULL,&routine,NULL);

pthread\_join(t1,NULL);

pthread\_join(t2,NULL);

}



3(a).write a c program to simulate INTER PROCESS COMMUNICATION (IPC) TECHNIQUE shared memory

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/sh

m.h>

#include <string.h>

int main() {

int i,shmid;

void\*shared\_memory;

char buff[100];

shmid=shmget((key\_t)2345,1024,0666|IPC\_CREAT); // id, storge, r/w permission

printf("KEY OF SHARED MEMORY is %d\n",shmid);

shared\_memory=shmat(shmid,NULL,0); // shared memory attached

printf("process attached at %p\n",shared\_memory);

printf("enter some data\n");

read(0,buff,100); //enter input from user

strcpy(shared\_memory,buff);

printf("you wrote %s\n",(char\*)shared\_memory);

printf("KEY OF SHARED MEMORY is %d\n",shmid);

printf("process is attached at %p\n",shared\_memory);

printf("data read from shared memory is %s\n",(char\*)shared\_memory);

return 0;

}

