1)

//Programmer Name: Sharvil Prabhudesai 20co41

//Program Title : Use of fork()

#include<stdio.h>

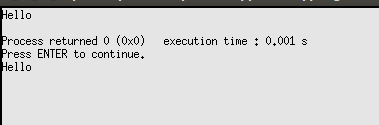
int main(){

fork();

printf("Hello \n");

return 0;

}



2)

//Programmer Name: Sharvil Prabhudesai 20co41

//Program Title : Use of multiple fork()

#include<stdio.h>

int main(){

fork();

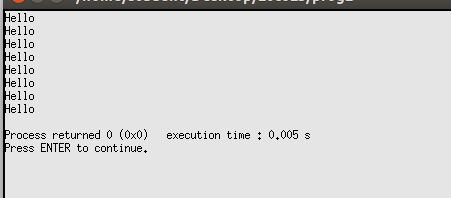
fork();

fork();

printf("Hello \n");

return 0;

}



3)

//Programmer Name : Sharvil Prabhudesai 20co41

//Program Title : Use of fork() and display whether it is a parent or child process

#include<stdio.h>

int main(){

int pid;

pid = fork();

if( pid < 0){

printf("Error : Cannot Process \n");

exit(1);

}

if( pid == 0){

printf("Child Process \n");

exit(0);

}

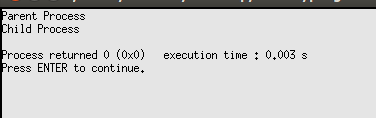
else {

printf("Parent Process \n");

}

return 0;

}



4)

//Programmer Name: Sharvil Prabhudesai 20co41

//Program Title : Use of fork() and sleep() and display whether it is a parent or child process

#include<stdio.h>

int main(){

int pid;

pid = fork();

if( pid < 0){

printf("Error : Cannot Process \n");

exit(1);

}

if( pid == 0){

sleep(2);

printf("I am a Child Process\n");

exit(0);

}

else {

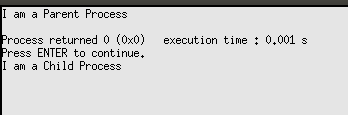
printf("I am a Parent Process \n");

exit(0);

}

return 0;

}



5)

//Programmer Name: Sharvil Prabhudesai 20co41

//Program Title : Use of fork() and sleep() and display whether it is a parent or child process along // with it’s PID

#include<stdio.h>

int main(){

int pid;

pid = fork();

if( pid < 0){

printf("Error : Cannot Process \n");

exit(1);

}

if( pid == 0){

printf("This is the child process. My pid is %d and my parent's id is %d \n", getpid(), getppid());

sleep(5);

printf("This is the child process. My pid is %d and my parent's id is %d\n", getpid(), getppid());

exit(0);

}

else {

sleep(10);

printf("This is the parent process. My pid is %d and my parent's id is %d.\n", getpid(), pid);

}

return 0;

}

