Processing Environment

Subject:- Unix Operating System

Class :- TYIT

Name	**PRN**
Dipankar Dubey	**2020BTEIT00057**

Assignment No - 1d

Title-Write the program to use wait/waitpid system call and explain what it do when call in parent.

Objectives –

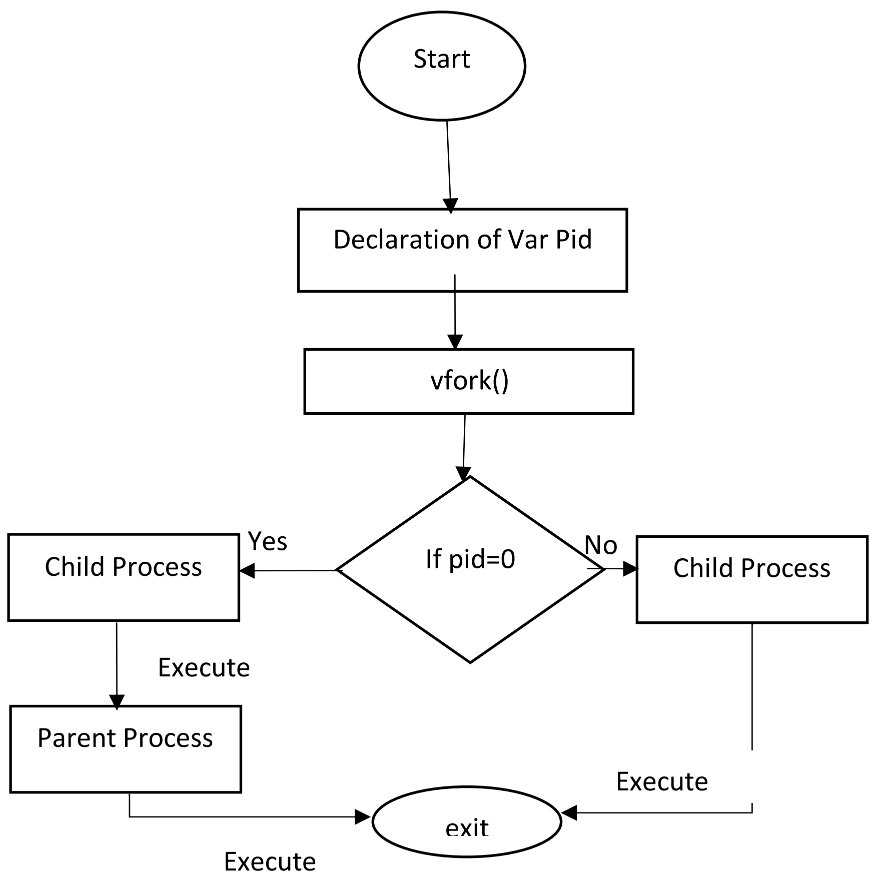
1. To learn about Processing Environment.
2. To know the difference between fork/vfork and various execs variations.
3. Use of system call to write effective programs.

Theory-

Difference between wait() and waitpid()

Wait()	Waitpid()
wait blocks the caller until a child process terminates	waitpid can be either blocking or nonblocking: • If options is 0, then it is blocking • If options is WNOHANG, then is it non-blocking
if more than one child is running then wait() returns the first time one of the parent's offspring exits	waitpid is more flexible: • If pid == -1, it waits for any child process. In this respect, waitpid is equivalent to wait • If pid > 0, it waits for the child whose process ID equals pid • If pid == 0, it waits for any child whose process group ID equals that of the calling process • If pid < -1, it waits for any child whose process group ID equals that absolute value of pid

Flowchart-



Program-

```
#include<stdlib.h>
#include<stdio.h>
#include<unistd.h>
#include<sys/wait.h>
void main()
{
pid_t id=fork();
if(id==0)
{
printf("Child Process Started..ProcessID = %d\n", getpid());
printf("In Child\n");
for(int i=0;i<5;i++)
{
printf("In Child : %d\n",i);
}
printf("Child Finished\n");
exit(0);
}
else
{
printf("Parent Process Started..ProcessID = %d\n", getpid());
printf("In Parent\n");
printf("Parent waiting\n");
wait(NULL);
printf("Parent Resumed\n");
for(int i=0;i<5;i++)
{
printf("In parent : %d\n",i);
}
printf("Parent Finished\n");
}
}
```

Output-

```
it@it-OptiPlex-3050:~/Desktop/57/UOS$ gedit 1d.c
it@it-OptiPlex-3050:~/Desktop/57/UOS$ gcc 1d.c
it@it-OptiPlex-3050:~/Desktop/57/UOS$ ./a.out
Parent Process Started..ProcessID = 11676
In Parent
Parent waiting
Child Process Started..ProcessID = 11677
In Child
In Child : 0
In Child : 1
In Child : 2
In Child : 3
In Child : 4
Child Finished
Parent Resumed
In parent : 0
In parent : 1
In parent : 2
In parent : 3
In parent : 4
Parent Finished
```

Conclusion:

The waitpid() call is more flexible than wait() system call as wait() would block the parent until child processes complete, while waitpid() can be implemented in blocking or unblocking ways

References:

www.tutorialspoint.com/unix_system_calls/