PES UNIVERSITY

Department of Computer Science & Engineering

OPERATING SYSTEMS

ASSIGNMENT:-1

SHELL PROGRAMMING

UE22CS242B

4th Semester, Academic Year 2023-2024

Date:01-02-2024

Q:1 ->

Write a program to create a child process which lists all the executing user processes. (Avoid creation of zombie process)

Code->

OS-assignment > C assg1.c > main()

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <stdlib.h>

int main()
{
    pid_t proc = fork();

    if (proc < 0)
    {
        printf("Error occured");
        exit(0);
    }
    else if (proc == 0)
    {
        printf("Child process id is %d \n", getpid());
        execlp("ps", "ps", "-u", NULL);
    }
    else
    {
        wait(NULL);
        printf("Parent process ID is %d\n", getpid());
    }
    return 0;
}
```

OUTPUT:-

```
(base) vvmohith@Mohiths-MacBook-Pro OS-assignment % cd "/Users/vvmohith/Desktop/OS/OS-assignm
ent/" && gcc assg1.c -o assg1 && "/Users/vvmohith/Desktop/OS/OS-assignment/"assg1
Child process id is 76441
ps: option requires an argument -- u
usage: ps [-AaCcEefhjlMmrSTvwXx] [-O fmt | -o fmt] [-G gid[,gid...]]
          [-g grp[,grp...]] [-u [uid,uid...]]
          [-p pid[,pid...]] [-t tty[,tty...]] [-U user[,user...]]
       ps [-L]
Parent process ID is 76440
```

Q-2:-

Create a global array with values [1, 6, 2, 4, 5, 8, 9, 0]. Sort the same within the child process, and display the values in the parent process. Are the displayed values in the sorted order? If not, why?

Code->

OS-assignment > assgn2.c > sorti(int *, int)

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <stdlib.h>

void sorti(int *arr, int len)
{
    int min = 0;
    int temp = 0;
    for (int i = 0; i < len - 1; i++)
    {
        min = arr[i];
        if (min > arr[i + 1])
        {
            temp = min;
            min = arr[i + 1];
            arr[i + 1] = temp;
        }
    }
}
void printarray(int *arrayy)
{
    for (int i = 0; i < 8; i++)
    {
        printf("%d ", arrayy[i]);
    }
}
```

```c
int main()
{
    int arr[8] = {1, 6, 2, 4, 5, 8, 9, 0};
    pid_t proc = fork();
    if (proc < 0)
    {
        printf("Error occured \n");
        exit(0);
    }

    else if (proc == 0)
    {
        printf("Inside child process with ID : %d\n ", getpid());
        sorti(&arr, 8);
        printarray(&arr);
    }

    else
    {
        wait(NULL);
        printf("The array is as follows \n");
        printarray(&arr);
    }

    return 0;
}
```

OUTPUT :

```
Inside child process with ID : 76909
 1 6 6 6 6 8 9 9 The array is as follows
1 6 2 4 5 8 9 0 %
(base) vvmohith@Mohiths-MacBook-Pro OS-assignment %
```

The answer is NO

REASON -> In a UNIX like Environment when a child process is forked from a parent process, the child process gets a copy-on-write snapshot of the parent's memory
Changes made by one process do not affect the other until they modify the shared data Hence it can lead to issues hen sorting

an array in a child process and expecting parent process to see the sorted result

Q-3:-Write a program which accepts two integers x and y. Now use exec to execute another user defined program that prints the product of x and y.

Code:-

OS-assignment > assgn4.c > main(int, char * [])

```c
// multiply.c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 3) {
        printf("There's a error ");
        exit(EXIT_FAILURE);
    }

    int x = atoi(argv[1]);
    int y = atoi(argv[2]);

    printf("Child process: x = %d, y = %d\n", x, y);
    printf("Product: %d\n", x * y);

    return 0;
}
```

(char [31])"Child process: x = %d, y = %d\n"

OS-assignment > assgn3.c > main(int, char * [])

```c
// main.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
    if (argc != 3) {
        printf("Error occured");
        exit(EXIT_FAILURE);
    }

    int x = atoi(argv[1]);
    int y = atoi(argv[2]);

    printf("Parent process: x = %d, y = %d\n", x, y);

    // Execute the second program using exec
    execl("./multiply", "multiply", argv[1], argv[2], NULL);

    // If execl fails, print an error message
    perror("execl failed");
    exit(EXIT_FAILURE);
}
```

OUTPUT ->

```
(base) vvmohith@Mohiths-MacBook-Pro OS-assignment % gcc assgn3.c -o main
(base) vvmohith@Mohiths-MacBook-Pro OS-assignment % gcc assgn4.c -o multiply
(base) vvmohith@Mohiths-MacBook-Pro OS-assignment % ./main 4 9
Parent process: x = 4, y = 9
Child process: x = 4, y = 9
Product: 36
```

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.

- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature:

Name:V V Mohith

SRN:PES2UG22CS641

Section: K

Date:01-02-2024