

PES UNIVERSITY

OS ASSIGNMENT-1

Name : Shivi Jain

SRN : PES2UG22CS526

SEC : I

1. Write a program to create a child process which lists all the executing user processes.(Avoid Creation of Zombie Process).

CODE:

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

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

    if (pid < 0) {
        fprintf(stderr, "Fork failed\n");
        return 1;
    } else if (pid == 0) {
        // Child process
        execlp("ps", "ps", "u", NULL);
    } else {
        // Parent process
        wait(NULL); // Wait for child to complete
    }

    return 0;
}
```

OUTPUT :

```
[01/31/24]seed@VM:~/.../Shivi$ gcc l.c
[01/31/24]seed@VM:~/.../Shivi$ ./a.out
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
seed      7818  0.0  0.0   6980  4444 pts/3    Ss   10:34   0:00 bash
seed      7841  0.0  0.0   4936  1528 pts/3    S+   10:35   0:00 ./a.out
seed      7842  0.0  0.0  10244  4784 pts/3    R+   10:35   0:00 ps u
[01/31/24]seed@VM:~/.../Shivi$
```

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 parent process. Are the displayed value in the sorted order ? if not why?

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
int global_array[] = {1, 6, 2, 4, 5, 8, 9, 0};
int array_size = sizeof(global_array) / sizeof(global_array[0]);

void bubbleSort(int arr[], int n) {
    int i, j;
    for (i = 0; i < n-1; i++) {
        for (j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

int main() {
    pid_t pid = fork();
    if (pid < 0) {
        fprintf(stderr, "Fork failed\n");
        return 1;
    } else if (pid == 0) {
        bubbleSort(global_array, array_size);
        printf("Child process sorted the array\n");
    } else {
        wait(NULL);
        printf("Parent process displaying sorted array:\n");
        for (int i = 0; i < array_size; i++) {
            printf("%d ", global_array[i]);
        }
        printf("\n");
        return 0;
    }
}
```

OUTPUT :

```
[01/31/24]seed@VM:~/.../Shivi$ gcc 2.c
[01/31/24]seed@VM:~/.../Shivi$ ./a.out
Child process sorted the array
Parent process displaying sorted array:
1 6 2 4 5 8 9 0
```

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:

3.c

```
3.c x
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main() {
    int x, y;
    printf("Enter two integers: ");
    scanf("%d %d", &x, &y);

    char x_str[10], y_str[10];
    sprintf(x_str, "%d", x);
    sprintf(y_str, "%d", y);
    execl("./3b", "3b", x_str, y_str, NULL);
    perror("execl failed");
    return 1;
}
```

3b.c

```
3.c x 3b.c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 3) {
        fprintf(stderr, "Invalid Arguments! Syntax: %s <x> <y>\n", argv[0]);
        return 1;
    }

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

    printf("Product of %d and %d: %d\n", x, y, x * y);

    return 0;
}
```

OUTPUT :

```
[01/31/24]seed@VM:~/.../Shivi$ gcc 3.c -o 3
[01/31/24]seed@VM:~/.../Shivi$ gcc 3b.c -o 3b
[01/31/24]seed@VM:~/.../Shivi$ ./3
Enter two integers: 7 5
Product of 7 and 5: 35
[01/31/24]seed@VM:~/.../Shivi$
```