PES UNIVERSITY

Department of Computer Science & Engineering

Operating Systems Assignment

UE23CS242B

Exercise 2 Submission

Name of the Student	Pranav Hemanth
SRN	PES1UG23CS433
Section	G
Department	CSE
Campus	RR

Please look at page 4 for execution

Jan -May 2025 Assignment SUBMISSION_**UE23CS242B**

Department of Computer Science & Engineering
Operating Systems Assignment

UE23CS242B

Q) Write a C program to do the following:
- Main program receives a string as runtime argument
- Create a child process to reverse the string
- The parent program should send signal to pause the child process
- The parent program should send a signal to continue the child process
- The child process should print the reversed string

Program screenshot:

Assignments > Exercise2 > C prog-exercise_2.c > main(int, char * [])

```c
// Main program receives a string as runtime argument
// Create a child process to reverse the string
// The parent program should send signal to pause the child process
// The parent program should send a signal to continue the child process
// The child process should print the reversed string

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

void reverse_and_print(const char *str);

int main(int argc, char *argv[])
{
    if (argc != 2)
    {
        fprintf(stderr, "Usage: %s <string_to_reverse>\n", argv[0]);
        return EXIT_FAILURE;
    }

    pid_t pid = fork();

    if (pid < 0)
    {
        perror("fork failed");
        return EXIT_FAILURE;
    }

```

```c

    if (pid == 0)
    {
        printf("Child process (PID: %d) started and waiting to be resumed...\n", getpid());

        // A delay to prevent SIGSTOP and SIGCONT from running at once
        for (int i = 5; i > 0; i--)
        {
            printf("Child preparing in %d...\n", i);
            sleep(1);
        }

        reverse_and_print(argv[1]);
        exit(EXIT_SUCCESS);
    }
    else
    {
        sleep(2);
        printf("Parent pausing child process (PID: %d)\n", pid);
        kill(pid, SIGSTOP);

        sleep(3);
        printf("Parent resuming child process (PID: %d)\n", pid);
        kill(pid, SIGCONT);

        wait(NULL);
        printf("Child process finished.\n");
    }

    return EXIT_SUCCESS;
}

void reverse_and_print(const char *str)
{
    int len = strlen(str);
    printf("Reversed string: ");
    for (int i = len - 1; i >= 0; i--)
    {
        putchar(str[i]);
    }
    putchar('\n');
}
```

Execution screenshot:

```
.venvpranavhemanth@Pranavs-MacBook-Pro-M3 Exercise2 %gcc prog-exercise_2.c -o prog-exercise_2
.venvpranavhemanth@Pranavs-MacBook-Pro-M3 Exercise2 %./prog-exercise_2
Usage: ./prog-exercise_2 <string_to_reverse>
.venvpranavhemanth@Pranavs-MacBook-Pro-M3 Exercise2 %./prog-exercise_2 "operating systems"
Child process (PID: 85729) started and waiting to be resumed...
Child preparing in 5...
Child preparing in 4...
Child preparing in 3...
Parent pausing child process (PID: 85729)
Parent resuming child process (PID: 85729)
Child preparing in 2...
Child preparing in 1...
Reversed string: smetsys gnitarepo
Child process finished.
.venvpranavhemanth@Pranavs-MacBook-Pro-M3 Exercise2 %
```