**AMRITA VISHWA VIDYAPEETHAM**

**BANGALORE CAMPUS- 560035**

******

19EAC211 – OPERATING SYSTEM

2022-2023

**OPERATING SYSTEM PROJECT**

**NAME and REG NO.:** Janani J – BL.EN.U4EAC21026

Ramu Keerthana P – BL.EN.U4EAC21055

**DEPARTMENT:** Electronics and Computer Engineering

**SECTION:** ’I’

**SEMESTER:** IV Semester

**Aim:**

Build a C program that can perform process tracing, tracking and control (implementation of

process diagram).

**Theory:**

1. Define the Process Diagram Structure: Decide on the representation of your process diagram. It could be a flowchart, a directed graph, or any other suitable structure that captures the sequence and dependencies of process steps.
2. Data Modelling: Identify the data elements involved in the process and create appropriate data models or classes to represent them. For example, you might have entities like "Process Step," "Resource," "Input Data," "Output Data," etc.
3. User Interface: Design a user interface where users can interact with the program. This interface should allow users to view and modify the process diagram, track the progress, and control the execution of the process.
4. Implement Process Tracing: Develop the functionality to trace the progress of the process. This involves keeping track of the current step, identifying completed steps, and managing the flow through the process diagram. You can use techniques like recursion or iterative algorithms to traverse the diagram.
5. Process Tracking: Implement the tracking mechanism to monitor the execution of each process step. This may involve logging timestamps, resource utilization, input/output data, and any other relevant information for each step.
6. Control Mechanisms: Provide controls to start, pause, resume, and stop the process execution. These controls should be able to handle both manual and automated execution of the process.
7. Error Handling: Implement error handling mechanisms to deal with exceptions or errors that may occur during the process execution. This includes validating inputs, handling runtime errors, and providing appropriate error messages or recovery options.
8. Integration: If required, integrate your program with external systems or tools. For example, you might need to connect with databases, APIs, or other software components to fetch or update data during the process.
9. Testing and Debugging: Thoroughly test your program to ensure it functions correctly. Debug any issues that arise during testing and make necessary adjustments.
10. Deployment and Maintenance: Once your program is tested and working as expected, deploy it in the desired environment. Maintain the program by fixing bugs, adding new features, and addressing user feedback.

The specific implementation details will depend on the programming language and technologies you choose. Make sure to break down the problem into smaller modules or functions for better code organization and readability.

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

int main() {

pid\_t pid;

// Fork a child process

pid = fork();

if (pid < 0) {

// Forking failed

fprintf(stderr, "Forking process failed.\n");

return 1;

} else if (pid == 0) {

// Child process

printf("Child process ID: %d\n", getpid());

printf("Parent process ID: %d\n", getppid());

printf("Executing child process...\n");

// Perform some task in the child process

// ...

sleep(5); // Simulating some work

printf("Child process completed.\n");

exit(0);

} else {

// Parent process

printf("Parent process ID: %d\n", getpid());

printf("Child process ID: %d\n", pid);

printf("Waiting for child process to complete...\n");

// Wait for the child process to finish

int status;

waitpid(pid, &status, 0);

if (WIFEXITED(status)) {

printf("Child process finished successfully.\n");

} else {

printf("Child process terminated abnormally.\n");

}

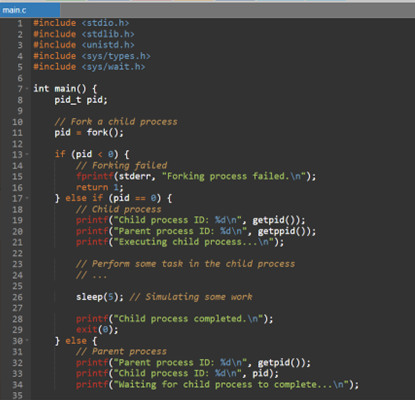
printf("Parent process completed.\n");

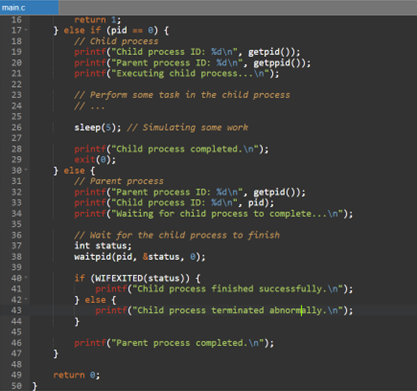
}

return 0;

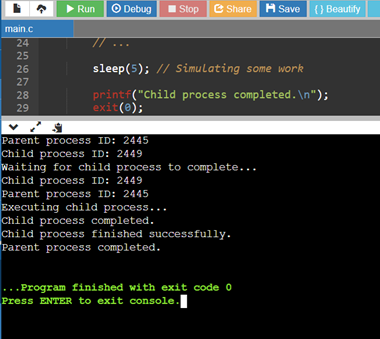
}

**Screenshot of the code:**

****

****

**Output:**

****

**Result:**

Therefore, the program which perform process tracing, tracking and control using process is executed.