CSA 0407- OPERATING SYSTEMS

LAB PRATICAL: **1**

NEW PROCESS AND DISPLAY PID AND PPID USING C PROGRAM

CODE:

```
#include <windows.h>
#include <stdio.h>
#include <tchar.h>

int main() {
    STARTUPINFO si;
    PROCESS_INFORMATION pi;
    ZeroMemory(&si, sizeof(si));
    si.cb = sizeof(si);
    ZeroMemory(&pi, sizeof(pi));

    // Display current (parent) process info
    DWORD parentPID = GetCurrentProcessId();
    printf("Parent Process:\n");
    printf("PID  = %lu\n", parentPID);

    // Path to the child program (for demonstration, we’ll relaunch the same program)
    TCHAR szCmdLine[] = _T("child.exe");

    // Create the child process
```

```
BOOL success = CreateProcess(
    NULL,           // Application name
    szCmdLine,      // Command line
    NULL,           // Process handle not inheritable
    NULL,           // Thread handle not inheritable
    FALSE,          // Set handle inheritance to FALSE
    0,              // No creation flags
    NULL,           // Use parent's environment block
    NULL,           // Use parent's starting directory
    &si,            // Pointer to STARTUPINFO structure
    &pi             // Pointer to PROCESS_INFORMATION structure
);

if (!success) {
    printf("CreateProcess failed (%lu).\n", GetLastError());
    return 1;
}

// Display child process info
printf("\nChild Process:\n");
printf("PID  = %lu\n", pi.dwProcessId);
printf("Parent PID (via current process) = %lu\n", parentPID);

// Wait until child process exits
WaitForSingleObject(pi.hProcess, INFINITE);

// Close process and thread handles
CloseHandle(pi.hProcess);
```

```
    CloseHandle(pi.hThread);

    return 0;
}
```