

Name: Rounak Sarkar
Roll No.: 21CS8118
Assignment: Q-A2

A2) Write a code such that takes input 'n' and then the parent process (p) creates n number of child processes (c1, c2, c3,.....cn) all of them as its direct descendent i.e. p is the parent of c1, c2, c3 and also cn. Each process including the parent and child processes does the following:

(a) display the pid & ppid once in the terminal and also store the values with timestamp in a common file (log.txt)

(b) sleep for 1 minute (this is just to allow the process tree to be visualized)

Hint: The parent will create the file log.txt and all its child processes can get access it. Get a snapshot of the process tree using forest option in ps command (check the pdf material provided). Provide a snapshot of the file log.txt

Ans:

```
#include <iostream>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
#include <fstream>
#include <ctime>

using namespace std;

void log_pid_ppid(int pid, int ppid, ofstream& log_file){
    time_t now = time(0);
    char* dt = ctime(&now);
    log_file << "PID: " << pid << ", PPID: " << ppid << ", Timestamp:"<< dt;
    cout << "PID:" << pid << ", PPID: " << ppid << endl;
}

int main(){
    int n;
    cout<< "Enter the number of child processes you want to create: " << endl;
    cin >> n;

    ofstream log_file("log.txt");
    log_file << "Parent process created" << endl;
    if(!log_file.is_open()){
        cerr << "Error: Couldn't open the log file" << endl;
        return 1;
    }
}
```

```

pid_t pid;

for (int i=0; i<n; i++){
    pid = fork();

    if (pid == -1){
        cout << " Error in creating child process" << endl;
        return 1;
    }

    else if (pid == 0){
        log_pid_ppid(getpid(), getppid(), log_file);
        sleep(60);
        return 0;
    }

    else{
        log_pid_ppid(getpid(), getppid(), log_file);
    }
}

for (int i=0; i<n; i++){
    wait(NULL);
}

log_file.close();

return 0;
}

```