COMP 322/L—Introduction to Operating Systems and System Architecture
Assignment #1--Process Creation Hierarchy

Objective:

To simulate process creation and destruction when implemented with linked lists.

Specification:

The program creates/destroys child processes based on choosing from a menu of choices, where each choice calls the appropriate procedure, where the choices are:

1) Initialize process hierarchy
2) Create a new child process
3) Destroy all descendants of a parent process
4) Quit program and free memory

Assignment:

- Create a process creation hierarchy as an array of length **MAX_PROCESSES** which references process control blocks (PCBs), indexed 0 to **MAX_PROCESSES-1.**
- Each PCB is a structure consisting of two fields:
  - parent: a PCB index corresponding to the process' creator
  - children: a pointer to a linked list, where each node contains the PCB index of one child process and a link to the next child in the linked list

- The necessary functions are simplified as follows:
  - **create_child()** represents the create function, which prompts for the parent process **p**. The function creates a new child process **q** of process **p** by performing the following tasks:
    - allocate memory for an unused **PCB[q]**
    - record the parent's index, **p**, in **PCB[q]**
    - initialize the list of children of **PCB[q]** as empty (NULL)
    - create a new link containing the child's index **q** and append the link to the children field of **PCB[p]**
  - **destroy_descendants()** represents the destroy function, which prompts for the parent process **p**. The function recursively destroys all descendent processes (child, grandchild, etc.) of process **p** by performing the following tasks: for each element **q** on the linked list of children of **p**:
    - **destroy_desecndants(q)** (recursively destroy all descendants of **q**)
    - free memory utilized by **PCB[q]** and set it to NULL
    - Free memory utilized by the node with id **q** and set it to NULL

What NOT to do:

- Do NOT modify the choice values (1, 2, 3, 4) or input characters and then try to convert them to integers--the test script used for grading your assignment will not work correctly.
- Do NOT turn in an alternate version of the assignment downloaded from the Internet (coursehero, chegg, reddit, github, ChatGPT, etc.) or submitted from you or another student from a previous semester—the test cases from this semester will not work on a previous semester's assignment.
- Do NOT turn in your assignment coded in another programming language (C++, C#, Java).

What to turn in:

- The source code as a C file uploaded to Canvas by the deadline of 11:59pm PST -Please check the syllabus for the late submission policy. 1-minute late counts as a day late, 1-day and 1-minute late counts as 2 days late, etc.)
- As a note, even though your code may compile on a compiler you have installed on your computer, I do not have access to your computer. I will be using the following free online compiler for testing, so make sure your code compiles with the following online C compiler before submitting: https://www.onlinegdb.com/online_c_compiler
  If it does not compile with the above compiler, the default grade is 0 points since I cannot run it.

Sample Output:

```
Process creation and destruction
--------------------------------
1) Initialize process hierarchy
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 1

Process list:
Process id: 0
        No parent process
        No child processes

Process creation and destruction
--------------------------------
1) Initialize process hierarchy
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 2
Enter the parent process id: 0

Process list:
Process id: 0
        No parent process
        Child process: 1
Process id: 1
        Parent process: 0
        No child processes

Process creation and destruction
--------------------------------
1) Initialize process hierarchy
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 2
Enter the parent process id: 0

Process list:
Process id: 0
        No parent process
        Child process: 1
        Child process: 2
Process id: 1
        Parent process: 0
        No child processes
Process id: 2
        Parent process: 0
        No child processes

Process creation and destruction
--------------------------------
1) Initialize process hierarchy
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 2
Enter the parent process id: 2

Process list:
Process id: 0
        No parent process
```

```
        Child process: 1
        Child process: 2
Process id: 1
        Parent process: 0
        No child processes
Process id: 2
        Parent process: 0
        Child process: 3
Process id: 3
        Parent process: 2
        No child processes

Process creation and destruction
--------------------------------
1) Initialize process hierarchy
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 2
Enter the parent process id: 0

Process list:
Process id: 0
        No parent process
        Child process: 1
        Child process: 2
        Child process: 4
Process id: 1
        Parent process: 0
        No child processes
Process id: 2
        Parent process: 0
        Child process: 3
Process id: 3
        Parent process: 2
        No child processes
Process id: 4
        Parent process: 0
        No child processes

Process creation and destruction
--------------------------------
1) Initialize process hierarchy
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 3
Enter the parent process whose descendants are to be destroyed: 2

Process list:
Process id: 0
        No parent process
        Child process: 1
        Child process: 2
        Child process: 4
Process id: 1
        Parent process: 0
        No child processes
Process id: 2
        Parent process: 0
        No child processes
Process id: 4
        Parent process: 0
        No child processes

Process creation and destruction
--------------------------------
1) Initialize process hierarchy
```

```
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 3
Enter the parent process whose descendants are to be destroyed: 0

Process list:
Process id: 0
        No parent process
        No child processes

Process creation and destruction
--------------------------------
1) Initialize process hierarchy
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 4

Quitting program...
```