

**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) Enter parameters
- 2) Create a new child process
- 3) Destroy all descendants of a process
- 4) Quit program and free memory

**Assignment:**

- Create a process creation hierarchy as a dynamic array of length  $n$  which references the process control blocks (PCBs), indexed 0 to  $n-1$
- 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 list
- The necessary functions are simplified as follows:
  - **create()** represents the create function, which prompts for the parent process PCB[p]. The function creates a new child process PCB[q] of process PCB[p] by performing the following tasks:
    - allocate a free 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 linked list of PCB[p]
  - **destroy()** represents the destroy function, which prompts for the parent process PCB[p]. The function recursively destroys all descendent processes (child, grandchild, etc.) of process PCB[p] by performing the following tasks:
    - for each element q on the linked list of children of PCB[p]
      - destroy(q) /\* recursively destroy all descendants \*/
      - free PCB[q]
      - deallocate the element q from the linked list

**What NOT to do (any violation will result in an automatic score of 0 on the assignment):**

- 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 alternative version of the assignment downloaded from the Internet (coursehero, chegg, reddit, github, etc.)
- Do NOT use any self-created or external libraries that cannot be located/utilized by zyLabs
- Do NOT turn in your assignment coded in another programming language (C++, C#, Java, Python, Perl, etc.)—it will NOT compile under zyLabs C compiler.

**What to turn in:**

The source code as a C file (**comp322\_asmt1.c**) uploaded to zyLabs by the deadline of 11:59pm PST (-20% per consecutive day for late submissions, up to the 4<sup>th</sup> day—note 1 minute late counts as a day late, 1 day and 1 minute late counts as 2 days late, etc.)

### Sample test run:

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

Enter selection: 1
Enter maximum number of processes: 5

Process creation and destruction
-----
1) Enter parameters
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 index: 0
cr[0]      /*creates 1st child of PCB[0] at PCB[1]*/

Process creation and destruction
-----
1) Enter parameters
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 index: 0
cr[0]      /*creates 2nd child of PCB[0] at PCB[2]*/

Process creation and destruction
-----
1) Enter parameters
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 index: 2
cr[2]      /*creates 1st child of PCB[2] at PCB[3]*/

Process creation and destruction
-----
1) Enter parameters
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 index: 0
cr[0]      /*creates 3rd child of PCB[0] at PCB[4]*/

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

Enter selection: 3
Enter the process whose descendants are to be destroyed: 0
de[0]      /* destroys all descendants of PCB[0] which includes:
PCB[4] PCB[3] PCB[2] PCB[1] */

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

Enter selection: 4
Quitting program
```