# **CSCI 331 Project One**

Note: This is a project from your zybook textbook.

## **Project objective**

Compare the performance of process creation and destruction when implemented with and without linked lists.

## **Description**

Version 1 of the process creation hierarchy uses linked lists to keep track of child processes as described in section "The process control block", subsection "The PCB data structure".

For the purposes of performance evaluation, the PCBs are simplified as follows:

* All PCBs are implemented as an array of size n.
* Each process is referred to by the PCB index, 0 through n-1.
* Each PCB is a structure consisting of only the two fields:
  + parent: a PCB index corresponding to the process's creator
  + children: a pointer to a linked list, where each list element contains the PCB index of one child process

The necessary functions are simplified as follows:

* create(p) represents the create function executed by 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
  + create a new link containing the child's index q and appends the link to the linked list of PCB[p]
* destroy(p) represents the destroy function executed by 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 progenies \*/
    - free PCB[q]
    - deallocate the element q from the linked list

Version 2 of the same process creation hierarchy uses no linked lists. Instead, each PCB contains the 4 integer fields parent, first\_child, younger\_sibling, and older\_sibling, as described in the subsection "Avoiding linked lists".

## **Assignment**

1. Implement the two versions of the process creation hierarchy.
2. Assume that PCB[0] is the only currently existing process and write a test program that performs a series of process creations and destructions. Ex:

cr[0] /\* creates 1st child of PCB[0] at PCB[1]\*/

cr[0] /\* creates 2nd child of PCB[0] at PCB[2]\*/

cr[2] /\* creates 1st child of PCB[2] at PCB[3] \*/

cr[0] /\* creates 3rd child of PCB[0] at PCB[4] \*/

de[0] /\* destroys all descendents of PCB[0], which includes processes PCB[1] through PCB[4] \*/

1. Run the test program repeatedly in a long loop using both versions and compare the running times to see how much time is saved in version 2, which avoids dynamic memory management.