Course COMP-8567
Assignment 02
Winter 2023
Due Date: Mar/01/2023
50 Marks
PART A: 25 Marks
PART B: 25 Marks

Part A

Write a C program that searches for a particular process in the process tree (rooted at a specified process) and outputs the requested information based on the input parameters.

Synopsis :

prctree [*root_process]* [*process_id*] [*OPTION*]

- Lists the PID and PPID of *process_id* if it belongs to the process tree rooted at *root_process*
    - ***root_process*** is the PID of a process that is a descendant of a current BASH process.
    - ***process_id*** is the PID of a process that is a descendant of a current BASH process.

OPTION

- **- c** additionally lists the PIDs of all the child processes (immediate descendent/s) of *process_id*
- **- s** additionally lists the PID and PPID of all the sibling processes of *process_id*
- **- gp** additionally lists the PID of the grandparent of *process_id*
- **- gc** additonally lists the PIDs and PPIDs of all the grandchildren of *process_id*
- **- z** additionally prints if *process_id* is DEFUNCT/ NOT DEFUNCT
- **- zl** additionally lists the PIDs of all the child processes of *process_id* that are currently in the defunct state

Part B

Write a C program that searches for defunct process/es in a process tree rooted at a specified process and forcefully terminates the parent process/es based on the input parameters.

Synopsis:

ztree [*root_process*] [*OPTION1*] [*OPTION2*]

- Forcefully terminates all the parent processes (except BASH) of defunct processes that belong to the process tree rooted at *root_process* and prints the PIDs of the terminated process/es
- *root_process* is the PID of a process that is a descendant of a current bash process.

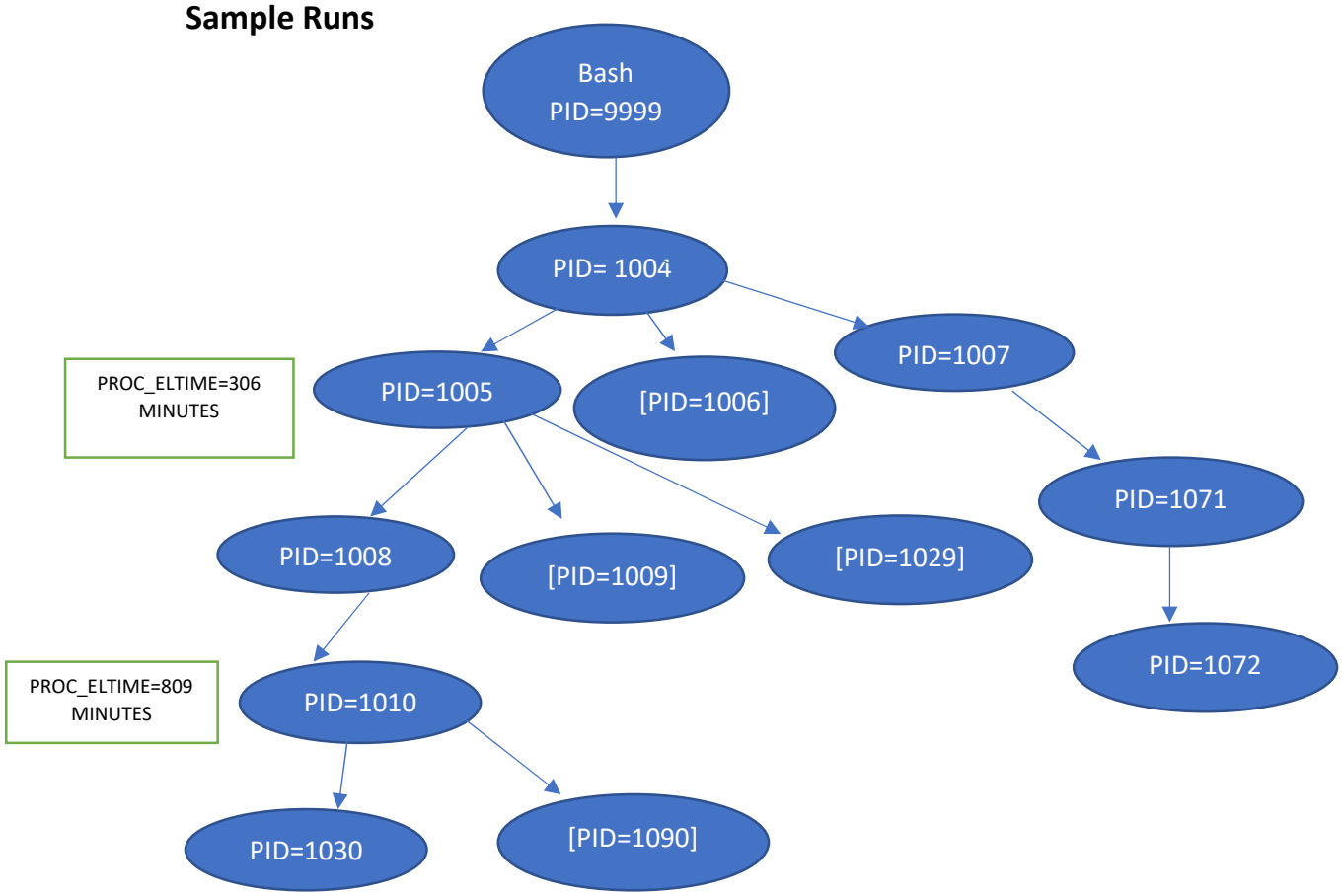
OPTION1

- **- t** forcefully terminates parent processes (whose elapsed time is greater than ***PROC_ELTIME***) of all the defunct processes in the process tree rooted at *root_process*
- **- b** forcefully terminates all the processes in the process tree rooted at root_process that have >= ***NO_OF_DFCS*** defunct child processes.

OPTION2

- **PROC_ELTIME** The elapsed time of the process in minutes since it was created (>=1)
- **NO_OF_DFCS** The number of defunct child processes (>=1)

Sample Runs



Note: In the above example, [PID=1006], [PID=1009], [PID=1029] and [PID=1090] are defunct (zombie) processes at the time of execution of the following programs

$ prctree 1004 1009
1009 1005
$prctree 1009 1004
//No output
$ prctree 1005 1010
1010 1008
$ prctree 1005 1020
//No output
 	
$ prctree 1005 1008 -gc
1008 1005
1030 1010
1090 1010
$ prctree 1004 1005 -zl
1005 1004
1009
1029
	$ prctree 1004 1029 -z
	1029 1005
	DEFUNCT
	$ prctree 1006 1040 -zl
	// No output
	$ prctree 1004 1008 -s
	1008 1005
	1009
	1029
	
	$ ztree 1005
	1005
	1010
	//Forcefully terminates 1005 and 1010
	$ ztree 1010
	1010
	//Forcefully terminates 1010
		$ ztree 1007
		//No process is forcefully terminated
		$ ztree 1005 -b 2
		1005
		//1005 is forcefully terminated, 1010 is not
		$ ztree 1004 -t 400
		1010
		// 1010 is forcefully terminated, 1005 is not
		
		$ prctree 1004 1004
		1004 9999
		$ prctee 1004 1004 -gc
		1004 9999
		1008 1005
		1009 1005
		1029 1005
		1071 1007

Submission:

You need to submit two files:

- prctree.c
- ztree.c