【2021 Network System Programming Homework 3】

Rules:

1. Please use C language in this homework and run your program on Ubuntu 20.04.
2. Please provide Makefile to compile your homework.
3. Do not copy the others homework definitely.
4. If you have any question, please send email to sp_ta@net.nsysu.edu.tw or drop by Room EC5018. However, TA will not help you to debug program.

Turn in your homework:

1. Please compress your homework into zip archive.
2. Naming rules: “SP_HW3.zip”.
3. Upload your homework (zip file) to NSYSU Cyber University (網路大學).
4. Deadline: 2021/10/26 14:10. You cannot get any credit if you do not turn in your homework before the deadline.

Part 1:

Request:

Write a program that lists the porcess ID and command name for all processes being run by the user named in the program's command-line argument.

You may find the userIdFromName() function from Listing 8-1, on page 159, useful. This can be done by inspecting the Name: and Uid: lines of all of the /proc/PID/status files on the system. Walking through all of the /proc/PID directories on the system requires the use of readdir(3), which is described in Section 18.8. Make sure your porgram correctly handles the possibility that a /proc/PID directory disappears between the time that the porgram determines that the directory exists and the time that it tries to open the corresponding /proc/PID/status file.

Part 2:

Request:

Write a program that draws a tree showing the hierarchical parent-child relationships of all processes on the system, going all the way back to init.

For each process, the program should display the process ID and the command being executed. The output of the program should be similar to that produced by pstree(1), although it does need not to be as sophisticated. The parent of each process on the system can be found by inspecing the PPid: line of all of the /proc/PID/status files on the system.

Part 3:

Request:

Write a program that lists all processes that have a particular file pathname open.

This can be achieved by inspecting the contents of all of the /proc/PID/fd/* symbolic links. This will require nested loops employing readdir(3) to scan all /proc/PID directories, and then the contents of all /proc/PID/fd entries within each /proc/PID directory. To read the contents of a /proc/PID/fd/n symbolic link requires the use of readlink(), described in Section 18.5.