Operating Systems

Lab 9 Exercise – Recap threads, signals and bash

Exercise 1

Given a text file, passed as an argument of the command line
Implement a concurrent program using three threads (T1, T2, T3) that process the file content in pipeline
T1: Read from file the next character
T2: Transforms the character read by T1 in uppercase
T3: Displays the character produced by T2
Upgrade the solution given in the thread slides, and the corresponding C program, to process correctly the first two characters of the file, and files with less than two characters.

Exercise 2

Write a **program** that is able to handle signals **`SIGUSR1, SIGUSR2`** and **`SIGALRM`** as follows:
Every 5 seconds it controls the sequence of signals received, and

- outputs an **error** message if it received signals **`SIGUSR1`** followed by signal **`SIGUSR2`** or viceversa,
- outputs a **success** message if **two** successive **`SIGUSR1`** (or **`SIGUSR2`**) signals,
- terminates if it has received at least **three** successive **`SIGUSR1`** (or **`SIGUSR2`**) signals.

Exercise 3

Write a **bash** script that runs the program of exercise **n. 2** to tests its correctness.

Exercise 4

A Web Server keeps an access log file with this format:

```
   178.1.192.33 goodguy [10/07/2015:13:55:36] GET index.html 200
   34.52.1.33 badguy [10/07/2015:14:04:10] POST /services/ask 200
   34.52.1.37 badguy [10/07/2015:14:04:10] POST /intranet/login 401
   34.52.177.48 okguy [10/07/2015:14:32:00] POST /intranet/login 401
   178.1.192.41 nastyguy [10/07/2015:18:29:01] POST /intranet/login 401
   178.1.192.42 nastyguy [10/07/2015:18:56:01] POST /intranet/login 200
   178.1.192.32 goodguy [11/07/2015:15:21:43] POST /intranet/login 200
   123.154.48.1 worstguy [11/07/2015:00:21:32] GET /services/list 200
```

Every file line indicates: the host IP address issuing the access request, the username of the user issuing the request, the date and time of the request, the request type (**`GET`** or **`POST`**), the name of the resource, and the server answer (**`200`** for “access granted”, **`401`** for “access denied”).
Write a **bash** script **`check.sh`** that takes as its argument a log file, a date, and a list of usernames.
Example:

```
> check.sh file.log 10/07/2015 badguy nastyguy worstguy
```

The script must output the list of the IP addresses from which one of the users of the given list has performed a **`POST`** access request on the given date.
The script must not output duplicate line information.
For the example file and command, the output is:

```
   34.52.1.33
   178.1.192.42
```