

## Lista 4 (Lab) Termin wysłania na SVN do 10.11.2019

- (10pt) Napisz skrypt w Bashu, który co sekundę pokazuje następujące informacje o systemie
  - Aktualną i średnią prędkość przesyłania danych po interfejsach sieciowych (odczytaj i zinterpretuj `/proc/net/dev` oraz wyświetl w B, KB lub MB w zależności od aktualnej prędkości)
  - Jak długo system jest uruchomiony w dniach, godzinach, minutach i sekundach (`/proc/uptime`)
  - Aktualny stan baterii w procentach (`/sys/class/power_supply/BAT0/uevent`)
  - Obciążenie systemu `/proc/loadavg`

Skrypt prezentuje "graficznie" historię poprzednich pomiarów np. prosty wykres słupkowy. Przykładowe programy z wykresami `s-tui` lub `bmon`. Można wykorzystać inne znaki w UTF-8. Zobacz też informacje o komendzie `tput` np. [link](#) i/lub [link](#).

- (10pt) Napisz skrypt w Bashu, który pokazuje informacje o wszystkich procesach (podobne jak program `ps`). Wykorzystaj system plików `procfs` (standardowo w systemie Linux montowanym w katalogu `/proc`) do pobrania informacji o procesach np. `cat /proc/1/status` wyświetla informacje o procesie 1. Dodatkowo wyświetl informację ile proces ma otwartych plików.
- (10pt) Napisz skrypt w Bashu wykorzystujący `REST API` do pobierania danych z dwóch przykładowych źródeł np. [Internet Chuck Norris Database \(ICNDb\)](#) i [The Cat API](#). Do zapytań RESTowych wykorzystaj `curl` lub `wget` do parsowania JSONa wystarczy program `jq` (`pacman -S jq`), dla osób korzystających z formatu XML dostępny jest program `xmllint` (`pacman -S libxml2`). Po uruchomieniu skryptu na ekranie pokaż obraz z bazy 'The Cat API' wykorzystując np. program `img2txt` (`pacman -S libcac`) lub `cating` oraz poniżej wyświetl losowy cytat z bazy Chucka Norrisa.
- (10pt) Napisz skrypt w Bashu, który przechowuje historię zmian podanej strony z internetu i powiadamia użytkownika w momentach zmian na stronie (do powiadamiania można wykorzystać komendę `xmessage`, `zenity` lub `echo` z terminala i wyświetlić zmiany). Przykładowe wywołanie programu

Terminal

```
$ ./webtrace.sh "http://ki.pwr.edu.pl" 600
```

gdzie pierwszym parametrem jest adres śledzonej strony, a drugim parametrem jest czas w sekundach co jaki strona będzie pobierana i sprawdzana czy nastąpiła zmiana, czyli w powyższym przykładzie strona `http://ki.pwr.edu.pl` będzie sprawdzana co 10 minut.

Wskazówka: Do pobrania strony można wykorzystać np. `lynx -dump http://ki.pwr.edu.pl` (man `lynx`, instalacja `pacman -S lynx`) lub `links` (man `links`, instalacja `pacman -S links`) lub `w3m` (man `w3m`, instalacja `pacman -S w3m`) lub ... . Do sprawdzania zmian na stronie: `diff` (man `diff`), ... . Do pamiętania historii można wykorzystać repozytorium `git`, instalacja `pacman -S git`.

- (10pt)\* Celem tego zadania jest dokładniejsze poznanie `ANSI escape sequences` i obsługi terminali. Bazując np. na grze `arkanoid` napisz prostą grę w tenisa `pong`. Zrób tak aby gra dostosowywała się do wielkości terminala. Przetestuj program na różnych emulatorach terminali np. `xterm`, `urxvt`, `termite`, `st`, `gnome-terminal`, `linux console` itp., czy są jakieś różnice?