

### Lista 3 (Lab) Termin wysłania na SVN do 27.10.2019

Na SVN proszę wysłać tylko zadania 6 i 7. Resztę zadań należy oddać na pierwszych (lub drugich) zajęciach po terminie 27.10.2019.

1. (5pt) Wytlumacz jakie pliki zawierają katalogi `/dev` oraz `/proc`. Wykorzystując polecenie `dd` odczytaj pierwszy sektor z dysku głównego (uwaga na prawa dostępu) lub podpiętego pendrive'a i wyświetl przez `hexdump -C`. Z katalogu `proc` wyświetl informacje o pamięci, procesorze i partycjach.
2. (5pt) Zapoznaj się z programem `ps` (`man ps`). Naucz się posługiwać tym programem, aby umieć sprawdzać co najmniej istnienie i parametry wybranych procesów (PID procesu i rodzica, stan procesu, priorytet, nice, ilość pamięci, zużycie czasu procesora). Uruchom też kilka terminali pokaż jakie urządzenie tty wykorzystują. Wykonując komendę `ps axjf` pokaż wszystkie procesy które podpięte są do tych terminali (kolumna TTY).
3. (5pt) Zapoznaj się z kompilatorem języka C (polecenie `gcc`) oraz języka C++ (polecenie `g++`). Uruchom poniższy program w języku C.

```
Terminal
$ cat > test.c
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("Hello, World!\n");
    return 0;
}
^D
$ gcc -Wall -pedantic test.c
$ ./a.out
```

Wytlumacz każdy z powyższych kroków. Co oznaczają opcje `-Wall` oraz `-pedantic`? Zobacz `man gcc`.

4. (5pt) Pokaż na przykładzie (np. `sleep 1000`, `sleep 1001`, ...) zarządzanie zadaniami wykorzystując `<polecenie> &` - uruchamianie w tle (background) oraz `jobs`, `fg`, `bg`, `kill` oraz `^Z`. Uruchom kilka zadań w tle i pokaż jak można nimi zarządzać, czyli zatrzymywać, wznowiać oraz kończyć ich działanie. Pokaż jak uruchomione zadanie (nie w tle), można w czasie działania uruchomić w tle np. wykonując komendę `sleep 100` (bez `&`) w czasie działania przełącz je do działania w tle.
5. (5pt) Poleceniem `mkfifo` (`man mkfifo`) utwórz potok nazwany (ang. *named FIFO*). Wywołując polecenie `cat` w różnych terminalach spowoduj wpisywanie danych do potoku przez jeden(ne) proces(y), i odczytywanie i wyświetlanie ich przez inne. Zaobserwuj, kiedy polecenie `cat` czytające z potoku czeka na więcej danych, a kiedy kończy pracę. Analogicznie, kiedy czeka na więcej danych (z klawiatury) polecenie `cat` piszące do potoku?
6. (8pt) Napisz program w języku C, który wykorzystując sekwencje Esc (ang. *escape sequences*) standardu ANSI wyświetli na ekranie napis "Hello, World!", po kolei we wszystkich dostępnych przez terminal kolorach. Czy terminal może wyświetlić 256 kolorów?
7. (10pt) Napisz potok poleceń (w skrypcie), który zamienia wszystkie nazwy plików w danym katalogu (bez podkatalogów) na małe litery, czyli wszystkie duże litery występujące w nazwach plików zostaną zamienione na małe, a małe litery pozostają oczywiście dalej małe. Skrypt powinien działać poprawnie na takich nazwach plików jak " ABC DEF", "-AbC aBc" i "--ABC DEF". Podane nazwy plików proszę zakładać w konsoli korzystając z polecenia `touch`.