

# Domaća zadaća - 05

Osnove korištenja operacijskog sustava Linux

14. siječnja 2018.

Za **svaki** zadatak treba napisati bash skriptu s rješenjem zadatka. Jedna se točka zadatka **može riješiti s više redova i naredbi** u skripti.

## Zadatak 1

- Ispišite **sve** procese koje je pokrenuo trenutni korisnik (i **samo** njih). Kao i uvijek, učinite to dinamički<sup>1</sup>, a ne hardkodiranjem svog korisničkog imena.
- Sortirajte prethodni ispis po zauzeću memorije.
- Ispišite stablo procesa za trenutnog korisnika.

Sljedeći se zadaci odnose na skriptu `even_bash.sh`. To je skripta koja prima jedan cjelobrojni argument, a ispisuje je li on paran ili neparan.

## Zadatak 2

- Koristeći ljsku `bash` i tu skriptu, provjerite parnost brojeva 27 i 42. Skriptu pokrenite sekvencom `./`, a ne naredbom `bash`.
- Nakon toga, pokrenite neku ljsku koja nije `bash` (npr. `sh`, koja je dostupna na većini sustava - jednostavno pokretanjem naredbe `sh`). Iz nove ljske, pokušajte na isti način pokrenuti skriptu. U komentar napišite u čemu je problem.

Sljedeći se zadatak odnosi na C program `signal_handler.c`.

## Zadatak 3

### 1. Signali SIGTERM vs. SIGKILL

- Prevedite taj program (npr. `gcc signal_handler.c`) i pokrenite ga.
- Pronađite PID tog programa. Dinamičko rješenje nije nužno, ali bit će dodatno nagrađeno (jer se rezultat koristi u sljedećoj točki). U svakom slučaju, postupak opišite u komentarima.
- Pokušajte "ubiti" taj proces naredbom `kill <pid>`. Zašto niste uspjeli?

---

<sup>1</sup>Ovisno o pristupu koji odaberete, ako vam je korisničko ime dulje od 8 znakova, ono bi moglo biti skraćeno na format "prvih\_7\_znakova+". U tom slučaju nađite način da doskočite tome.

- Pokušajte ga sada ubiti slanjem adekvatnog signala. Zašto ste sad uspjeli, iako je naoko situacija u kôdu ista?

## 2. Exit codes

- Pokrenite program i pošaljite mu SIGUSR1.
- Pokrenite program i pošaljite mu SIGINT. Ponovno pokrenite program i pošaljite mu SIGUSR2.
- Istražite što u ljusti radi sekvenca `$?` i potom u komentaru opišite kako možemo (osim čitanjem ispisane izlazne poruke) iz ljuste utvrditi je li izvođenje programa uspjelo ili ne. Misli se na semantiku izvođenja, u smislu "je li obavljeno što je trebalo biti obavljeno".

## Zadatak 4

Skripta priložena uz treću laboratorijsku vježbu, `lab03-3.sh`, sadrži beskonačnu petlju. Zadaci u nastavku odnose se na nju.

- Pokrenite proces u *foregroundu* te ga zatim pošaljite u pozadinu. Objasnite kako ste to napravili. Kako biste proces poslali u pozadinu odmah pri pokretanju?
- Aktivnom procesu (status *RUNNING*) pošaljite SIGSTOP. Kako ste to napravili? Što se dogodilo s procesom? Može li se signal SIGSTOP programski obraditi?
- Nastavite izvođenje procesa u pozadini. Kako ste to izveli?
- Ugasite terminal. Objasnite što se dogodilo s procesom. Koji signal mu je poslan? Navedite barem jedan način kako biste pokrenuli proces i osigurali nastavak njegova izvođenja nakon gašenja terminala.