

# Operativni sistemi Septembar 2 26.09.2017.

Napraviti u `/home/ispit1` direktorijum u skladu sa indeksom i asistentom kod koga slusate kurs. Na primer, student koji slusa kurs kod Vlade, sa indeksom 101/2014, treba da napravi folder `v_mi14101_s2`, a student sa indeksom 12/2014 koji slusa kurs kod Ognjena treba da napravi folder `o_mi15012_s2`. Za svaki zadatak napraviti odgovarajuci `.c` fajl unutar ovog foldera (1.c, 2.c ... 5.c).

Ispit se radi 3h. Svaki zadatak nosi po **20%** tj. **6 poena**. Na izlaz za greske mozete ispisivati sta god zelite. Strogo se drzite navedenih formata ispisa za standardni izlaz!

**Zabranjeno je koristiti `system()` funkciju! Kod obavezno prevoditi sa opcijom `-std=c99`! Ukoliko zadatak zahteva dinamicku alokaciju, koriscenje staticke povelaci 0 poena!**

1. Napisati program koji kao argumente komandne linije prima broj niti `n` koje treba da pokrene i putanju do fajla u kome se nalazi sekvenca nukleinske kiseline (mala slova '`a`', '`t`', '`c`', '`g`' u proizvoljnom redosledu bez belina izmedju; primer sekvence: `'aatctatatggata'`). Ako je velicina fajla `M` bajtova (karaktera), svaka nit obradjuje `M/n` bajtova (pretpostaviti da `n` deli `M`), dakle nit sa indeksom 0 karaktere `[0, M/n)`, nit sa indeksom 1 `[M/n, 2*M/n)` itd. Niti u svom delu fajla trazе stop kodone, tj. niske karaktera '`tag`', '`taa`' ili '`tga`', i za svako pojavljivanje jedne od ovih niski dodaju poziciju pocetka u **globalni niz**. Ispisati globalni niz iz `main()` funkcije po zavrsetku obrade. Pretpostaviti da nece biti vise od 2048 stop kodona i da stop kodon ne moze da se desi na granici obrade izmedju 2 niti. Primer pokretanja (ako broj argumenata nije dobar ili fajl ne postoji `exit code` treba da bude 1):

```
$ ./1 2 genom.txt      | sadrzaj fajla 'genom.txt': aatagggatcccgatgagtcgataatc (podvuceno nit 0)
2 14 22                | zbog konkurentnosti moze se dobiti drugaciji redosled ispisa (npr. 14 2 22)
```

2. Napisati program koji kao argument komandne linije prima putanju do fajla. Program treba da cita rec po rec (maksimalna duzina reci 256) i ukoliko je procitana rec **pozitivan cetvorocifreni broj**, da se potradi da ga zakljuca za citanje. Ukoliko je broj ne moze da se zakljuca, program normalno nastavlja dalje sa obradom. Ukoliko je uspesno zakljuacan zameniti to pojavljivanje broja u fajlu sa `####`. Proveriti broj argumenata komandne linije i da li fajl postoji ( `exit code` 1 ako nesto od toga nije ispunjeno). Primer pokretanja:

Pokretanje:	<code>./2 primer.txt</code>	<code>./2</code>	<code>./2 ne_postoji.txt</code>
primer.txt (pre):	Brojevi 1421 i 7234 su zakljucani, 12345 i -4000 se ne racunaju, a 7891 nije zakljucan...	----	-----
Katanci:	R(start=8, duzina=5), W(start=16, duzina=4)	----	-----
primer.txt (posle):	Brojevi 1421 i 7234 su zakljucani, 12345 i -4000 se ne racunaju, a #### nije zakljucan...	----	-----
Exit code:	0	1	1

3. Napisati program koji u petlji ucitava sa standardnog ulaza **liniju po liniju**. U svakoj liniji se nalaze 2 broja `x` i `y` i karakter `op` koji moze biti '+', '-', '\*' ili '/'. Ucitava se sve do kraja ulaza (`Ctrl+D`). Za svako ucitavanje program pokrece **ново дете процес** i u njemu izvrsava komandu: `'expr x op y'` (npr. za `x=5`,

y=7 i op=+ komanda je: 'expr 5 + 7'). Rezultat komande **expr** roditelj ispisuje na standardni izlaz (u detetu preusmeriti STDOUT pre exec poziva). Primer pokretanja:

```
$. /3
7 6 +      | ← unos, prva linija
13         | ← ispis iz roditelja
3 -4 *     | ← unos druga linija
-12        | ← ispis iz roditelja (nakon ovoga Ctrl+D za kraj programa)
```

4. Napisati program koji kao argument komandne linije prima broj sekundi od Epohe i ispisuje naziv meseca na srpskom jeziku za prosledjeni broj sekundi. Naziv ispisati malim slovima. Primer pokretanja:

```
$. /4 1504969947
septembar
```

5. Napisati program koji kao argument komandne linije prima putanju do direktorijuma i pokrece komandnu **ls -l** **putanja** u dete procesu, preusmerava standardni izlaz te komande u *PIPE*, filtrira ovaj sadrzaj tako da ispisuje samo osmu kolonu izlaza ove komande (**zabranjena staticka alokacija bafera za liniju, ne znate duzinu najduze reci u liniji, ne morate proveravati broj argumenata komandne linije**).

```
$. /5 Desktop
```

#### Izlaz iz ls -l:

```
total 18656
drwxr-xr-x 2 korisnik korisnik 4096 Dec 16 10:08 9
drwx----- 7 korisnik korisnik 4096 Dec 3 10:22 MATF2016jhp
-rw-r--r-- 1 korisnik korisnik 18104 Dec 9 08:29 MatfPlusPlus.ods
-rw-r--r-- 1 korisnik korisnik 41390 Dec 7 23:58 MatfPlusPlusOgranicenjaV2.zip
```

#### Očekivani izlaz iz programa

```
10:08
10:22
08:29
23:58
```

## POSIX niti - dodatak

Sve funkcije za rad sa POSIX nitima vraćaju pozitivnu vrednost koda greške ako je do greske doslo, a nulu inace. Zadaci koji koriste ove funkcije se moraju linkovati sa **-lpthread**. Potpisi najbitnijih funkcija slede:

```
int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
                   void *(*start_routine) (void *), void *arg);
int pthread_join(pthread_t thread, void **retval);
int pthread_mutex_init(pthread_mutex_t *mutex, const
                      pthread_mutexattr_t *attr);
int pthread_mutex_destroy(pthread_mutex_t *mutex);
int pthread_mutex_lock(pthread_mutex_t *mutex);
int pthread_mutex_unlock(pthread_mutex_t *mutex);
```