

# Operativni sistemi Jun 1 19.06.2018.

Napraviti u `/home/ispit` direktorijum u skladu sa indeksom, semestrom i asistentom kod koga slusate kurs. Na primer, student sa indeksom 101/2015 koji slusa kurs kod Vladimira Kuzmanovica u letnjem semestru treba da napravi folder **2\_mi15101\_v**, a student sa indeksom 12/2016 koji slusa kurs kod Ognjena Kocica u zimskom semestru treba da napravi folder **1\_mi16012\_o** (vodeci broj oznacava semestar). Za svaki zadatak napraviti odgovarajući `.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 greške možete ispisivati šta god želite. Strogo se držite navedenih formata ispisa za standardni izlaz!

## Kod prevoditi sa opcijom `-std=c99`. Zabranjeno koriscenje `system(3)` funkcije.

1. Korisnik kao argument komandne linije prosledjuje jedan ceo broj koji predstavlja broj dana za koliko treba promeniti datum kada se program pokrece. Napisati program koji pomera datum za onoliko dana za koliko korisnik zeli. Program treba da ispiše ime dana u nedelji koji odgovara novom datumu na srpskom jeziku malim slovima.

Pokretanje:	<code>./1 14</code>	<code>./1 10</code>	<code>./1</code>
Std. izlaz:	utorak	petak	-----
Exit code:	0	0	1

2. Napisati program koji pomocu niti izracunava histogram slova koja se nalaze u fajlu koji se programu prosledjuje kao argument komandne linije. Jedna nit broji pojavljivanja jednog slova. Dakle prva nit broji pojavljivanja slova 'a', druga slova 'b', treca slova 'c' itd. Niti ne razlikuju velika i mala slova (dakle pokrece se 26 niti). Sve karaktere koji nisu slova treba ignorisati. Na kraju rada, program stampa dobijeni histogram i slovo koje se najvise puta javilo u fajlu. Nije dozvoljeno naknadno utvrdjivanje maksimuma linearnom pretragom na osnovu zapamcenih brojeva pojavljivanja slova. Sinhronizaciju ostvariti pomocu mutex-a.

Pokretanje:	<code>./2 1.txt</code>	<code>./2</code>	<code>./2 ne_postoji.txt</code>
U fajlu:	Ana voli milovana	----	<u>Nema fajla</u>
Std. izlaz:	4 0 0 0 0 0 0 0 2 0 0 2 1 2 2 0 0 0 0 0 0 2 0 0 0 0 a	----	-----
Exit code:	0	1	1

3. Napisati program koji u dete procesu pokrece komandu **ls -l**. Roditeljski proces ceka na izvršenje komande i prikazuje **nazive** samo onih fajlova kod kojih ostali korisnici na sistemu imaju **rwX** privilegije. U slucaju da je izvršavanje komande neuspesno, roditeljski proces ispisuje poruku **neuspeh** i prekida svoje izvršavanje sa exit kodom koji je vratila komanda **ls**.
4. Program kao argument komandne linije dobija putanju do regularnog fajla. U tom fajlu, svaku otkljucanu rec program treba da zameni sa tom istom reci koja je obrnuta kao u

ogledalu. U slucaju da je rec vec zakljucana, program treba da je preskoci i pokusa da izmeni sledecu rec. Na kraju rada, program ispisuje broj uspesno izmenjenih reci i broj reci koje nije uspeo da izmeni. ***U tabeli reci napisane masnim slovima su zakljucane.***

Pokretanje:	./4 1.txt	./4 ne_postoji.txt	./4
Fajl pre:	Ovo je <b>malo</b> vise <b>reci i</b> karaktera <b>nego sto vam</b> treba	<u>Nema fajla</u>	---
Fajl posle:	ovO ej <b>malo</b> esiv <b>reci i</b> aretkarak <b>nego sto vam</b> abert	<u>Nema fajla</u>	---
Std. izlaz:	5 6	-----	---
Exit code:	0	1	1

5. Napisati program koji kao argumente komandne linije prima naziv objekta *deljene memorije*. Potrebno učitati strukturu:

```
typedef struct {
    sem_t ready;
    sem_t done;
    double points[MAX_SIZE];
    unsigned nPoints; /* broj tacaka */
};
```

Program ceka na semaforu **ready** da podaci budu spremni za obradu, pronalazi dve najblize tacke u nizu tacaka **points** i na kraju postavlja semafor **done**. Niz **points** se sastoji od alternirajucih **x** i **y** koordinata tacaka ( $x_0, y_0, x_1, y_1 \dots$ ). Na kraju rada, program stampa najkrace rastojanje koje je izracunao. Promenljiva **MAX\_SIZE** je 1024 (dakle najvise 512 tacaka). NAPOMENA: Linkovati sa **-lrt** i **-lpthread**.

Pokretanje:	./5 /inmem	./5	./5 /nepostoji
Niz <b>points</b> :	<u>-1.3, 1.7, 15.4, -101.3, -1.4, 1.9, 13.2, -99.5, -17, 0</u>	----	-----
Std. izlaz:	0.2236	----	-----
Exit kod:	0	1	1

## POSIX niti - dodatak

Sve funkcije za rad sa POSIX nitima vraćaju pozitivnu vrednost koda greske 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);
int pthread_cond_init(pthread_cond_t *cond, const pthread_condattr_t *attr);
int pthread_cond_destroy(pthread_cond_t *cond);
int pthread_cond_signal(pthread_cond_t *cond);
int pthread_cond_broadcast(pthread_cond_t *cond);
int pthread_cond_wait(pthread_cond_t *cond, pthread_mutex_t * mutex);
```