

Operativni sistemi Septembar 1 12.09.2017.

Napraviti u `/home/ispit1` direktorijum u skladu sa indeksom, semestrom i asistentom kod koga slusate kurs. Na primer, student koji slusa kurs u prvom semestru kod Vlade, sa indeksom 101/2015, treba da napravi folder `v1_mi15101_s1`, a student sa indeksom 12/2015 koji slusa kurs kod Ognjena u drugom semestru treba da napravi folder `o2_mi15012_s1`. 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 povlaci 0 poena!

1. Napisati program koji **u petlji** ucitava rec ne duzu od 63 karaktera sa standardnog ulaza. Nakon ucitavanja reci program ceka da mu se posalje signal SIGUSR1, SIGUSR2 ili SIGTERM (**cekanje na signal je u istoj petlji kao citanje stringa**). Ukoliko program primi SIGUSR1 na standardni izlaz ispisuje ucitanu rec naopako, ukoliko primi SIGUSR2 ispisuje ucitanu rec u kojoj su mala slova zamenjena velikim slovima, ukoliko primi SIGTERM program prekida sa izvršavanjem (*exit code* 0). **String obradljivati unutar petlje (u signal handler-u samo postaviti flag koji govori koja operacija treba da se izvrši)**. Primer pokretanja (**obavezno kompajliranje sa `-std=c99`**):

\$./1

dabar ← ovo je uneta rec za kojom sledi signal SIGUSR1

rabad ← ovo je ispis, uneta rec *dabar* je obrnuta zbog signala SIGUSR1

MiLiVoje ← ovo je naredna uneta rec za kojom sledi signal SIGUSR2

MILIVOJE ← ovo je ispis, uneta rec *MiLiVoje* je konvertovana u velika slova zbog SIGUSR2

← poslat SIGTERM, program ne ispisuje nista samo završava sa radom

2. Sa standardnog ulaza se ucitava broj cvorova grafa $|V|$, a zatim i graf predstavljen matricom povezanosti M dimezija $|V| \times |V|$. Ukoliko postoji grana od cvora i do cvora j onda je $M[i][j] = 1$, a nula inace. Pokrenuti $|V|$ niti i u svakoj izracunati ulazi stepen odgovarajuceg cvora (u nultoj niti za cvor $i=0$ itd.). Ispisati indeks cvora sa najvećim ulaznim stepenom iz *main* funkcije. Koristiti muteks za sinhronizaciju. Primer pokretanja:

\$./2

3 ← broj cvorova $|V|$

0 1 1

1 1 0 ← matrica povezanosti

0 1 0

1 ← cvor 1 ima najveći ulazni stepen (ulazni stepen za ovaj cvor je 3)

3. Napisati program koji kao argumente komandne linije prima *relativne* putanje do *FIFO* fajlova. Koriscenjem *epoll* API-ja program istovremeno motri ove fajlove cekajuci na dogadjaj citanja.

Program cita realne brojeve iz *FIFO* fajlova sve dok ima brojeva i ispisuje *relativnu* putanju do *FIFO* fajla ciji je zbir procitanih brojeva najveći. Primer pokretanja:

```
$ ./3 ../fajl0 a/b/fifo1 ./fifo2 | ../fajl0 | a/b/fifo1 | ./fifo2 | <- FIFO fajlovi
| 1.2 2.5 | 0.5 -1 3.2 | -1.3 | <- brojevi u fajlovima
../fajl0 <- ispisuje se ova putanja jer je 1.2+2.5=3.7 najveći od zbirova za FIFO fajlove
```

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

```
$ ./4 1504969947
subota
```

5. Napisati program koji kao argumente komandne linije prima ime objekta *deljene memorije*. Potrebno učitati strukturu (**ARRAY_MAX** je 1024):

```
typedef struct {
    sem_t inDataReady;
    int array[ARRAY_MAX];
    unsigned arrayLen;
} OsInputData;
```

i ispisati na **standardni izlaz** brojeve koji su stepeni **trojke**. Dodatno, pre bilo kakvog obrade, potrebno je sačekati na semafor *inDataReady* (pretpostaviti da je ispravno inicijalizovan).

<u>Pokretanje programa:</u>	./1 /inmem	./1	./1 /nepostoji	./1 /somemem
<u>Vrednosti array:</u>	3, 1, 2, 15, 81, 29	-----	-----	7, 27, 48, 258, 343
<u>Standardni izlaz:</u>	3 1 81	-----	-----	27 343
<u>Exit kod:</u>	0	1	1	0

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);
```