

Operativni sistemi Januar 2 31.01.2017.

Napraviti u `/home/ispit1` direktorijum u skladu sa indeksom i tokom kojem pripadate. Na primer, student prvog toka sa indeksom 101/2015 treba da napravi folder **1_mi15101**, a student drugog toka sa indeksom 12/2015 treba da napravi folder **2_mi15012**. 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!

1. Napisati program koji ispisuje ime dana u nedelji i redni broj dana u mesecu za datum koji se dobije kada se na datum pokretanja programa doda broj dana zadat kao argument komandne linije. Možete pretpostaviti da su argumenti komadne linije ispravni ukoliko ih ima.

Pokretanje programa:	<code>./1 8</code>	<code>./1</code>
Izlaz:	sreda 8	-----
Exit kod:	0	1

2. Napisati višenitni program koji računa maksimum svih zbirova kolona matrice. Kao argument komadne linije program prima putanju do fajla u kome se nalazi matrica. Prva dva broja matrice su broj redova N i broj kolona M , a zatim sledi $N*M$ celih brojeva. Potrebno je pokrenuti M niti i u svakoj niti izračunati sumu odgovarajuće kolone i ažurirati globalni maksimum. Koristiti muteks za sinhronizaciju. Ispisati globalni maksimum iz `main()` funkcije.
3. Napisati program koji u posebnim procesima vrši operacije nad niskama. Roditeljski proces sa standardnog ulaza u petlji učitava po reč i komandu (slovo). Dalje, za svako učitavanje roditelj pokreće dete proces i prosleđuje mu reč i komandu. Dete proces nakon što obradi reč šalje obrađenu reč roditelju koji je ispisuje na standardni izlaz. Reči nisu duže od 63 karaktera, a moguće komande su: **l** (prebaci sva slova u mala), **u** (prebaci sva slova u velika) i **r** (obrne nisku).

Pokretanje programa:	<code>./3</code>
Ulaz:	BaLkAn l katar r UgAnDa u
Izlaz:	balkan ratak UGANDA
Exit kod:	0

4. Napisati program koji kao argumente komandne linije prima putanju do objekta *deljene memorije* i broj. Potrebno učitati strukturu:

```
typedef struct {  
    sem_t inDataReady;
```

```

sem_t outDataReady;
char s[MAX_STR_LEN];
} OsInputData;

```

i šifrovati polje ove strukture s cikličnim dodavanjem broja koji je zadat kao argument komandne linije. Naravno, pre promene podataka potrebno je sačekati na semafor **inDataReady**, a nakon obrade postaviti semafor **outDataReady**. NAPOMENA: Linkovati sa **-lrt**, **MAX_STR_LEN** je 256.

Pokretanje programa:	./4 /inmem 1	./1	./1 /nepostoji
Vrednosti s (pre):	banana	-----	-----
Vrednost s (posle):	cbobob	-----	-----
Exit kod:	0	1	1

5. Program kao argumente komandne linije prima putanju do fajla i reč i treba da ispiše pozicije i tip katanca za sva zaključana pojavljivanja ove reči u fajlu (svaku u novom redu).

Pokretanje programa:	./5 1.txt bas	./5
Sadržaj fajla:	Ovo je bas , ali bas lako. Jednostavno bas !	-----
Izlaz:	7 w 39 r	-----
Exit kod:	0	1
Objasnjenje:	Prvo bas je zaključano F_WRLCK, a treće sa F_RDLCK katancom. Srednje bas nije zaključano.	Ako nema arg. ili fajla.

POSIX niti - dodatak

Sve funkcije za rad sa POSIX nitima vraćaju pozitivnu vrednost koda greške ako je do greške došlo, a nulu inače. 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);
```