

# Operativni sistemi Januar 1 16.01.2018.

Napraviti u **/home/ispit1** direktorijum u skladu sa indeksom, grupom i asistentom kod koga slusate kurs. Na primer, student grupe A sa indeksom 101/2016 koji slusa kurs kod Vladimira Kuzmanovica treba da napravi folder **a\_mi16101\_v**, a student grupe B sa indeksom 12/2016 koji slusa kurs kod Ognjena Kocica treba da napravi folder **b\_mi16012\_o**. 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. Napisati program koji za broj sekundi od Epohe, prosledjen kao argument komandne linije, ispisuje naziv meseca koji odgovara tom trenutku. Nazivi meseca su srpski nazivi koji pocinju sa malim slovom (npr. januar, februar, mart...).

Pokretanje:	./1 1515410719	./1 1520594719	./1
Std. izlaz:	januar	mart	-----
Exit code:	0	0	1

2. Napisati program koji kao argumente komandne linije prima putanju do fajla, sekvencu karaktera **W** i broj niti **K** koje treba pokrenuti. Za fajl velicine **N** svaka nit broji pojavljivanja **W** u jednoj sekciji fajla i dodaje taj broj na **globalnu sumu**. Nit sa indeksom 0 u sekciji  $[0, N/K)$ , nit sa indeksom 1 u sekciji  $[N/K, 2*N/K)$  itd. Iz **main funkcije** ispisati ukupan broj pojavljivanja **W** u fajlu. Pretpostaviti da se **W** ne moze naci na granici izmedju sekcija i da je **N** deljivo sa **K**.

Pokretanje:	./2 dir/1.txt peer2peer 3	./2	./2 ne_postoji.txt bla 20
Fajl:	sekvenca_peer2peer2peer_se_u_prvom_redu_nalazi_2x u drugom redu dva peer2peer \$\$\$\$ peer2peer \$\$\$\$ u?poslednjem?redu?nijednom#####	----	<u><b>Nema fajla</b></u>
Std. izlaz:	4	----	-----
Exit code:	0	1	1

3. Napisati program koji kao argument komandne linije prima putanju do fajla i jednu od niski: **'-w'**, **'-l'** ili **'-c'**. Program pokrece dete proces koje treba da izvrši komandu terminala **wc** za prosledjene argumente (npr. **wc -c dir/1.txt** za argumente **'dir/1.txt'** i **'-c'**). Preusmeriti standardni izlaz iz dete procesa, procitati ga u roditeljskom procesu i ispisati na standardni izlaz. Ukoliko dete proces ne završi uspesno, tj. *exit code* 0, ispisati **'Neuspeh'** iz roditelja.

Pokretanje:	./3 neki_direktorijum/jos_jedan/fajl.bla -w	./3 ne_postoji.txt -l	./3
Fajl:	Ovo su neke reci u fajlu...	<u><b>Nema fajla (wc ce exit 1)</b></u>	-----
Std. izlaz:	6	Neuspeh	-----
Exit code:	0	0	1

4. Napisati program koji kao argumente komandne linije prima putanje do FIFO fajlova. Program cita karaktere (bajtove) iz FIFO fajlova prateci sve fajlove istovremeno koriscenjem **poll** interfejsa. Ispisati **naziv** FIFO fajla iz kog je procitano **najvise** karaktera.

Pokretanje:	./4 /tmp/fifo /tmp/second_fifo dir/bla/third_fifo	./4 /tmp/f1 /tmp/ne_postoji	./4
FIFO 1:	Ovo su neki karakteri	-----	---
FIFO 2:	Ovo je malo vise karaktera	<b><u>Nema fajla</u></b>	---
FIFO 3:	Ovo je zaista najvise karaktera	-----	---
Std. izlaz:	third_fifo	-----	---
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 inDataReady;
    sem_t dataProcessed;
    char str[ARRAY_MAX];
} OsInputData;
```

i prebaciti sva mala slova niske *str* u velika i sva velika slova niske *str* u mala (invertovati *case*). Dodatno, pre bilo kakvog obrade, potrebno je **sacekati** na semafor *inDataReady*, nakon obrade **postaviti** semafor *dataProcessed* (pretpostaviti da je ispravno inicijalizovan). NAPOMENA: Linkovati sa **-lrt**.

Pokretanje programa:	./1 /inmem	./1	./1 /nepostoji	./1 /somemem
Niska <i>str</i> pre:	Bla_#!TraLa??Buc	-----	-----	NekaNISka
Niska <i>str</i> nakon:	bLA_#!tRAIA??bUC	-----	-----	nEKAnisKA
Exit kod:	0	1	1	0

## 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);
```