

# Operativni sistemi Jun 2 04.07.2017.

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 101/2015, treba da napravi folder `v_mi15101_j2`, a student sa indeksom 12/2015 koji slusa kurs kod Ognjena treba da napravi folder `o_mi15012_j2`. 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 koriscenje `system()` funkcije! Kod prevoditi sa `-std=c99` opcijom!**

1. Napisati program koji za trenutak pokretanja ispisuje na standardni izlaz koji je broj nedelje u godini (nedelje se broje pocevsi od prvog **ponedeljka** u godini). **Ispis je samo jedan broj.**
2. Napisati program koji u dete procesu pokrece komandu `find . -type f` i prosledjuje rezultat roditelju. Roditeljski proces treba da ispise **broj fajlova** cija ekstenzija odgovara ekstenziji koja se dobija kao argument komandne linije. Pokretanje: `./2.txt` . **Pomoc:** otvoriti `pipe` i izvršiti redirekciju u detetu tako da se `stdout` deteta preusmeri kroz `pipe` ka roditelju, a zatim nakon toga pozvati odgovarajuci `exec` u detetu; u roditelju citati iz `pipe`-a liniju po liniju.
3. Napisati program koji kao argument komandne linije prima putanju do fajla i dve reci. U navedenom fajlu program treba da zameni sva pojavljivanja prve reci sa drugom reci (pretpostaviti da su reci iste duzine). Pre nego sto prepise staru rec, program treba da je zakljuca. Ukoliko je druge proces vec zakljucio neko pojavljivanje reci, nastaviti normalno sa obradom bez zamene reci. Dodatno, potrebno je ispisati broj reci koje nisu zamenjene (zbog toga sto je drugi proces drzao katanac). Maksimalna duzina reci u fajlu je 256.

<b><u>Pokretanje programa:</u></b>	<code>./3 dira/1.txt</code> alati stati	<code>./3</code>	<code>./3</code> nepostoji ab ba
<b><u>Tekst u dira/1.txt:</u></b>	Ovo su alati koje treba zameniti. Dakle alati se menjaju.	----	-----
<b><u>Katanci na dira/1.txt:</u></b>	Read lock od 8. bajta duzine 5	----	-----
<b><u>Standardni izlaz:</u></b>	1	----	-----
<b><u>Sadrzaj dira/1.txt nakon izvorsavanja:</u></b>	Ovo su alati koje treba zameniti. Dakle stati se menjaju.	----	-----
<b><u>Exit kod:</u></b>	0	1	-----

4. Napisati program koji kao argumente komandne linije prima putanje do FIFO fajlova. Program istovremeno za sve fajlove ceka na dogadjaj da je kroz neki FIFO stigao podatak koriscenjem ***poll*** interfejsa. Kroz FIFO fajlove pristizu samo celi brojevi, cita se sve dok se ne zatvore svi FIFO fajlovi na strani koja pise u FIFO. Na standardni izlaz ispisati **maksimalni procitan broj** i **naziv** FIFO fajla (**ne putanju nego NAZIV**).

<b><u>Pokretanje programa:</u></b>	./4 /tmp/f0 /tmp/fl	./4	./4 ne_postoji /tmp/fifo
<b><u>Brojevi u /tmp/f0:</u></b>	1 2 3 4	-----	-----
<b><u>Brojevi u /tmp/fl:</u></b>	-5 2 3 -1 7 1	-----	-----
<b><u>Standardni izlaz:</u></b>	7 fl	-----	-----
<b><u>Exit kod:</u></b>	0	1	1

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 čiji binarni zapis ima **bar 4 jedinice** u sebi. Dodatno, pre bilo kakvog obrade, potrebno je sačekati na semafor *inDataReady* (pretpostaviti da je ispravno inicijalizovan). **Ne postavljati semafor nakon obrade!**

<b><u>Pokretanje programa:</u></b>	./1 /inmem	./1	./1 /nepostoji	./1 /somemem
<b><u>Vrednosti array:</u></b>	3, 1, 2, 15, 29	-----	-----	7, 120, 48, 258
<b><u>Standardni izlaz:</u></b>	15 29	-----	-----	120
<b><u>Exit kod:</u></b>	0	1	1	0