

Operativni sistemi kolokvijum

04.11.2017. grupa A

Napraviti u **/home/ispit1** direktorijum u skladu sa indeksom, grupom i asistentom kod koga slusa kurs. Na primer, student sa indeksom 101/2016 koji kurs slusa kod kolege Vladimira treba da napravi folder **v_mi16101_a**, a student sa indeksom 12/2016 koji kurs slusa kod kolege Ognjena treba da napravi folder **o_mi16012_a**. Za svaki zadatak napraviti odgovarajući .c fajl unutar ovog foldera (1.c, ... 5.c).

Kolokvijum se radi 2.5h. Svaki zadatak nosi po **20%** tj. **4 poena**. **Pazite na obradu grešaka, exit kod, curenje memorije, zatvaranje fajlova i zatvaranje direktorijuma – na to se takođe gube poeni. Na izlaz za greške možete ispisivati šta god želite. Strogo se držite navedenih formata ispisa za standardni izlaz!**

U svim zadacima je zabranjeno korišćenje system(3) funkcije. Kod kompajlirati sa -std=c99 opcijom.

- Programu se prosledjuju putanja do fajla i broj sekundi od epohe. Program treba da podesi vreme pristupa i modifikacije prosledjenom fajlu na prosledjeni broj sekundi. **POMOC:** Nakon pokretanja programa uradite na primer stat 1.txt i pogledajte datum koji se dobije za Access time i Modification time, treba da odgovara vremenu u redu Rezultujuca vremena:.

<u>Pokretanje programa:</u>	./1 1.txt 1509644770	./1	./1 ne_postoji 1509367190
<u>Rezultujuca vremena:</u>	2.11.2017. 18:46:10	----	-----
<u>Standardni izlaz:</u>	-----	----	-----
<u>Exit kod:</u>	0	1	1

- Napisati program koji prima **putanju do direktorijuma** i novi **naziv** za taj direktorijum. Program treba da **preimenuje** direktorijum u skladu sa drugim argumentom komandne linije. **NAPOMENE:** Proverite da li **putanja** pokazuje na direktorijum. Slobodno pretpostaviti da ne postoji fajl sa **nazivom** u koji treba preimenovati (na primer, u prvoj koloni tabele, **a/b/d** ne postoji pre pokretanja programa).

<u>Pokretanje programa:</u>	./2 a/b/c d	./2 dir novi	./2	./2 ne_postoji novi	./2 1.txt novi
<u>Putanja koja se dobije:</u>	a/b/d	novi	----	-----	-----
<u>Standardni izlaz:</u>	-----	-----	----	-----	-----
<u>Exit kod:</u>	0	0	1	1	1

3. Napisati program koji prima putanju do **fajla** i vrši zamenu prava pristupa za vlasnicku grupu i ostale korisnike.

Prava pristupa pre:	rw-rw-r--	rwXrwxr-x	Nepostojeci fajl	Pogresan br. arg.
Pokretanje programa:	./3 1.txt	./3 a.out	./3 ne_postoji	./3
Standardni izlaz:	-----	-----	-----	-----
Prava pristupa nakon:	rw-r--rw-	rwXr-xrwx	-----	-----
Exit kod:	0	0	1	1

4. Napisati program koji prima kao argumente komande linije putanje do 2 fajla. Prvi fajl u svakom redu sadrži jedan ceo broj i rec, npr. sadržaj bi mogao da izgleda ovako:

7 Milivoje
213 dzivdzan
153 bla

Broj predstavlja udaljenje od pocetka drugog fajla, a **rec** koja sledi je rec koju treba upisati u drugi fajl pocevši od pozicije zadate brojem. Dakle, rec *Milivoje* treba upisati u drugi fajl od pozicije 7 u tom fajlu, rec *dzivdzan* pocevši od 213 itd. Primer pokretanja:

./4 fajl_sa_recima fajl_koji_se_menja

Ukoliko jedan ili oba fajla ne postoje ili broj argumenata komadne linije nije odgovarajuci, završiti program sa exit code-om 1.

5. Napisati program koji prima putanju do direktorijuma i ispisuje razliku u velicini najveceg i najmanjeg regularnog fajla. Pretražuju se fajlovi koji se nalaze unutar zadatog direktorijuma rekurzivno (dakle i u poddirektorijumima itd.).

U slucaju da nije dobar broj argumenata komandne linije, da prosledjena putanja ne postoji ili da nije direktorijum (na primer prosledjen je regularan fajl) završiti sa exit code-om 1.

Zabranjeno je koriscenje ugradjene *nftw()* funkcije! Slobodno pretpostaviti da ako je zaista prosledjen direktorijum u njemu ima bar 1 regularni fajl!

Pokretanje: ./5 dirA

```
dirA size=4096
|-- 1.txt size=1017
|-- 2.c size=200
|-- dirB size=4096
    |-- OS.pdf size=2000000
    |-- OS.odp size=1500000
    |-- C size=4096
    |-- Ubuntu16.04.iso size=865075200
```

Standardni izlaz:

865075000