

1. Napisati program koji kao argumente komandne linije prima listu regularnih fajlova. Za svaki fajl pokrenuti posebnu nit u kojoj se taj fajl obrađuje (i -ta nit obrađuje i -ti fajl). Fajlovi sadrže samo znakove '+', '-' i beline. Svaka nit treba da izračuna skor za fajl, pri čemu se skor računa kao razlika broja '+' i '-' karaktera. Dodatno, svaka nit mora da ažurira globalni skor tako što će ga uvećati za izračunati lokalni skor (koristiti atomičke promenjive za sinhronizaciju tamo gde je to potrebno, muteksi su zabranjeni). U main() funkciji ispisati vrednost lokalnog skora za sve niti redom (vratiti ovu vrednost kao povratnu vrednost funkcije koju nit izvršava) i na kraju vrednost globalnog skora (svaki broj u svom redu).
2. Napisati program koji pokreće komandnu ls -l u dete procesu (fork()+exec()), preusmerava standardni izlaz te komande u PIPE, filtrira ovaj sadržaj tako da ispisuje samo petu kolonu izlaza ove komande (to je veličina fajla).

PRIMER:

```
Izlaz iz ls -l
drwxr-xr-x 2 korisnik korisnik 4096 Dec 16 10:08 9
drwx----- 7 korisnik korisnik 4096 Dec 3 10:22 MATF2016jhp
-rw-r--r-- 1 korisnik korisnik 18104 Dec 9 08:29 MatfPlusPlus.ods
-rw-r--r-- 1 korisnik korisnik 41390 Dec 7 23:58 MatfPlusPlusOgranicenjaV2.zip
-rw-r--r-- 1 korisnik korisnik 25801 Dec 7 23:56 MatfPlusPlus.pdf
Očekivani izlaz iz programa
4096
4096
18104
41390
25801
```

3. Napisati program sa više niti koji računa sumu matrice koja se dobija tačka po tačka proizvodom dve kvadratne matrice (eng. dot product). Kao argument komandne linije program prima putanju do fajla u kome se nalaze matrice. Prvi broj u ovom fajlu predstavlja dimenziju kvadratne matrice N, a zatim sledi 2*N 2 realnih brojeva koji su elementi matrica (prvih N 2 za prvu matricu, a drugih za drugu). Potrebno je pokrenuti N niti tako da jedna nit računa sumu tačka po tačka proizvoda za jedan par redova. Koristiti mutekse za sinhronizaciju. Ukupnu sumu ispisati iz main() funkcije. Primer tačka po tačka proizvoda:

```
1 2
3 4
2 2
X
2 2
=
2 4 => 1*2 2*2 => suma reda = 6
6 8 => 3*2 4*2 => suma reda = 14
```

4. Napisati program koji kao argument komandne linije prima putanju do fajla u kome je potrebno zameniti sva pojavljivanja karaktera a sa A. Program pokreće komadu terminala tr kojoj na standardni ulaz redirekcijom kroz PIPE prosleđuje sadržaj navedenog fajla. Takođe, program kroz drugi PIPE prihvata standardni izlaz pokrenute komande tr i ispisuje ga iz roditeljskog procesa. U slučaju da komanda tr završi greškom roditeljski proces ispisuje na standardni izlaz Neuspeh. Pomoć: Pokretanje komande tr iz terminala izgleda ovako:

```
$ echo "neki tekst" | tr e E
$ nEki tEkst
```

Dakle, ulaz se očekuje na standardni ulaz, pojavljivanja prvog karaktera se menjaju drugim.

5. Napisati program koji u beskonačnoj petlji učitava cele pozitivne brojeve. Nakon učitavanja broja program čeka na signal. Ukoliko program dobije signal SIGUSR1 ispisuje kvadrat broja, ukoliko dobije signal SIGUSR2 ispisuje kub broja, ako je u pitanju signal SIGINT ispisuje obrnut broj (npr. za 123 ispisuje se 321), a na SIGQUIT se izlazi iz programa bez obrade poslednje učitano broj. Osim brojeva ne ispisivati dodatan tekst. Smatrati da naredni signal neće biti poslat dok se ne završi obrada tekućeg broja.

6. Napisati program koji u fajlu koji se dobija kao argument komandne linije bezbedno menja sva pojavljivanja reči Milivoje sa Dragutin. Pod bezbednim menjanjem smatra se da se deo fajla gde je pronađena reč Milivoje zaključa za pisanje, zatim se promeni sadržaj fajla sa Milivoje na Dragutin i na kraju se otključa katanac (postupak ponoviti za svako pojavljivanje). Ukoliko nije moguće zaključati fajl, ispisati jedno slovo i dva broja: w ili r u zavisnosti od toga da li katanac koji neki drugi proces drži za pisanje ili čitanje, poziciju u fajlu gde počinje reč koju nije bilo moguće zaključati i identifikator procesa koji drži konfliktni katanac. Pomoć: Kada pročitate reč Milivoje koristite fseek() da se vratite unazad u fajlu.