

Jun 2015 Grupa A

1. Kao argument komandne linije dobija se putanja do direktorijuma. Potrebno je proći kroz sve fajlove koji se nalaze direktno u navedenom direktorijumu, i za svaki kreirati dete proces. Dete proces se odmah završava u slučaju da fajl nije regularan. Dete proverava da li je dužina fajla manja od 10 bajtova - koliko jeste kreira nisku „mali fajl: ***“ (***) zameniti sa sadržajem fajla), inače kreira nisku „veci fajl: +++“ (+++ zameniti sa prvih 10 karaktera fajla). Ova niska se preko pipe-a vraća roditeljskom procesu koji je ispisuje na standardni izlaz. Maksimalna dužina putanje do fajla je 1023 karaktera.
2. Kao argument komandne linije dobija se putanja do fajla. Potrebno je kreirati 10 niti i svakoj proslediti redom jednu od cifara 0, 1, 2, ..., 9. Svaka nit otvara fajl čije je ime učitano i računa koliko se puta prosleđena cifra pojavila u tom fajlu. Na kraju rada nit ispisuje koliko se ta cifra puta pojavila u fajlu i uvećava globalnu promenljivu koja sadrži ukupan broj pojavljivanja cifara u fajlu. Kada sve niti završe izvršavanje, glavni program ispisuje ukupan broj cifara u fajlu.

Grupa B

1. Program prima ime fajla kao argument komandne linije – u tom fajlu nalaze se imena direktorijuma i za svaki od njih je potrebno kreirati po jedan dete proces. Dete računa ukupnu veličinu svih regularnih fajlova kreiranih u zadnjih 60 minuta koji se nalaze u odgovarajućem direktorijumu (bez rekurzivnog obilaska hijerarhije) i tu informaciju preko pipe-a šalje roditeljskom procesu koji je ispisuje na standardni izlaz.
2. Kao argument komandne linije dobija se putanja do fajla. Potrebno je kreirati 26 niti i svakoj proslediti ime fajla, kao i jedno od slova a, b, c, ..., z. Svaka nit otvara datoteku i računa koliko se puta prosleđeno slovo pojavilo u njoj. Na kraju rada nit ispisuje koliko se to slovo puta pojavilo u datoteci i ukoliko je nit dobila samoglasnik, uvećava globalnu promenljivu koja sadrži ukupan broj malih samoglasnika u datoteci. Kada sve niti završe izvršavanje, glavni program ispisuje ukupan broj malih samoglasnika u datoteci.

Operativni sistemi, jul 2015

1. Napisati program koji kao argument komandne linije dobija putanju do direktorijuma i ceo pozitivan broj k . Za svaki regularni fajl koji se nalazi na većoj dubini od dubine k

dir_1/dir_2/.../dir_k/dir_{k+1}/.../dir_n/file_name

u odnosu na prosleđeni direktorijum se taj regularni fajl (korišćenjem eksterne komande *mv*) prebacuje u direktorijum na nivou k , tj. na lokaciju

dir_1/dir_2/.../dir_k/dir_{k+1}_dir_{k+2}_..._dir_n_file_name

a zatim se i briše sam direktorijum. Jedno pokretanje sa sadržajem hijerarhije pre i posle pokretanja može biti:

./1 a/ 1

a/ a1 b/ b1 b2 c/ c1 d/ d1	a/ a1 b_b1 b_b2 b_c_c1 d_d1
--	--

2. Napisati program koji kao argumente dobija nisku *file_name* i cele brojeve max , t i n . Program prvo kreira datoteku *file_name* veličine max , a zatim prolazi kroz petlju n puta, i u svakoj iteraciji petlje kreira dete proces. Između kreiranja svaka dva nova procesa roditelj se usplavljuje na 1 sekundu. Svaki

dete proces sa jednakim šansama bira da li da postavi katanac za čitanje ili katanac za pisanje na datoteku *file_name*. Oba katanca se postavljaju na slučajnom intervalu [*pocetak*, *kraj*) ($0 \leq \text{pocetak} < \text{kraj} \leq \text{max-1}$). Ukoliko je katanac uspešno postavljen izlazi poruka

Proces *pid*: Uspešno je postavljen read/write katanac na delu [*početak*, *kraj*) na *zzz* sekundi.
ili ako nije

Proces *pid*: Nije postavljen katanac read/write katanac na delu [*početak*, *kraj*).

Broj *zzz* označava slučajan broj sekundi iz intervala [$0, t$] i na taj broj sekundi se dete uspavljuje, nakon čega oslobađa katanac (ako ga je zauzeo) i gasi se proces. Roditelj tek posle kreiranja svih procesa čeka na njih. Za pokretanje programa: `./2 neki_fajl 30 5 9` izlaz bi mogao da bude

Proces 6148: Uspešno je postavljen write katanac na delu [3 14) na 2 sekundi.

Proces 6149: Uspešno je postavljen write katanac na delu [0 1) na 0 sekundi.

Proces 6150: Uspešno je postavljen read katanac na delu [4 22) na 2 sekundi.

Proces 6151: Nije postavljen katanac write na delu [7 12).

Proces 6152: Uspešno je postavljen read katanac na delu [22 23) na 3 sekundi.

Proces 6153: Uspešno je postavljen read katanac na delu [6 22) na 0 sekundi.

Proces 6154: Uspešno je postavljen read katanac na delu [8 12) na 2 sekundi.

Proces 6155: Uspešno je postavljen write katanac na delu [12 25) na 3 sekundi.

Proces 6156: Nije postavljen katanac read na delu [12 19)

Operativni sistemi, septembar 2015, 1. grupa

1. Kao argument komandne linije dobija se putanja do direktorijuma. Neophodno je *ručno implementiranim obilaskom hijerarhije* proći rekurzivno kroz sve fajlove koji se nalaze u navedenom direktorijumu. Za svaku kombinaciju prava pristupa potrebno je ispisati broj svih fajlova (uključujući i direktorijume) koji imaju tu kombinaciju prava, a kojima je pristupano 2015. godine. Ispis raditi samo za kombinacije za koje postoji barem jedan fajl sa tom kombinacijom. Na primer, rezultat pokretanja bi mogao da bude ispis

```
0644 : 20
0711 : 3
0755 : 7
0775 : 2
```

2. Kao argument komandne linije dobija se putanja do direktorijuma, karakter *c* i korisničko ime. Pomoću funkcije *nftw/ftw* proći kroz hijerarhiju datu putanjom i za svaku regularnu datoteku za koju važi da joj je vlasnik prosleđeni korisnik izračunati koliko se puta karakter *c* pojavio u njoj. Datoteku učitati u jedan bafer (pretpostaviti da ni jedna datoteka nije veća od 2048 bajtova). Ukupan broj pojavljivanja karaktera *c* u datoteci računati tako što se za svaki deo bafera veličine 10 bajtova kreira posebna nit, koja računa koliko se puta u tom delu datoteke pojavio karakter *c* i taj broj dodati u ukupnu sumu.

Nije potrebno zaključavati promenljive prilikom izmene, odnosno nije potrebno koristiti mutex-e.

Npr. za pokretanje programa

```
./prebroj_karaktere temp_dir/ a user1
```

ispis bi mogao da bude

```
temp_dir/a.txt 10
```

```
temp_dir/b.txt 3
```

```
temp_dir/dir2/c.txt 12
```

Operativni sistemi, septembar 2015, 2. grupa

1. Kao argument komandne linije dobija se putanja do direktorijuma. Neophodno je *ručno implementiranim obilaskom hijerarhije* proći rekurzivno kroz sve fajlove koji se nalaze u navedenom direktorijumu. Za **korisnička imena** na sistemu ispisati broj fajlova (uključujući i direktorijume) čiji je vlasnik svaki od korisnika, pri čemu se razmatraju samo fajlovi menjani 2015. godine. Ispis raditi samo za korisnike koji poseduju barem jedan fajl u direktorijumu. Na primer, rezultat pokretanja bi mogao da bude ispis

```
root 1  
ispit1 7
```

2. Kao argument komandne linije dobija se putanja do direktorijuma, broj n i oktalna konstanta koja označava prava pristupa. Pomoću funkcije *nftw/ftw* proći kroz hijerarhiju datu putanjom i za svaku regularnu datoteku za koju važi da su joj prava pristupa jednaka prosleđenoj konstanti izračunati koliko se puta karakter 'a' pojavio u njoj. Datoteku učitati u jedan bafer (pretpostaviti da ni jedna datoteka nije veća od 2048 bajtova). Ukupan broj pojavljivanja karaktera 'a' u datoteci računati tako što se za svaki deo bafera veličine n bajtova kreira posebna nit, koja računa koliko se puta u tom delu datoteke pojavio karakter 'a' i taj broj dodati u ukupnu sumu.

Nije potrebno zaključavati promenljive prilikom izmene, odnosno nije potrebno koristiti mutex-e.

Npr. za pokretanje programa

```
./prebroj_karaktere temp_dir/ a user1
```

ispis bi mogao da bude

```
temp_dir/a.txt 10
```

```
temp_dir/b.txt 3
```

```
temp_dir/dir2/c.txt 12
```