

Operativni sistemi kolokvijum

12.11.2016.

Napraviti u **/home/ispit1** direktorijum u skladu sa indeksom i tokom kojem pripada. Na primer, student prvog toka sa indeksom 101/2015 treba da napravi folder **1_mi15101**, a student drugog toka sa indeksom 12/2015 treba da napravi folder **2_mi15012**. Za svaki zadatak napraviti odgovarajući **.c** fajl unutar ovog foldera (1.c, 2.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!

1. Napisati funkciju `bool osIsPublicFile(const char *fpath);` koja vraća *true* ako ostali korisnici (eng. others) imaju rw prava pristupa nad *regularnim fajlom* na putanji 'fpath'. Testirati ovu funkciju pozivom iz *main*-a. Kao argument komandne linije programa prima se putanja do fajla. Pozvati `osIsPublicFile` funkciju i ispisati *true* ili *false* u zavisnosti od rezultata koji funkcija vrati.

| | | | | | |
|---------------------------------------|------------------|-------------------|-------------------|-------------------|--------------------|
| <u>Prava pristupa za fajl:</u> | 1.txt: -rw-rw-rw | a.out: -rwxr-xr-x | Nepostojeći fajl | Pogresan br. arg. | Pogresan tip fajla |
| <u>Pokretanje programa:</u> | ./1 1.txt | ./1 a.out | ./1 nepostoji.txt | ./1 | ./1 dir_a |
| <u>Standardni izlaz:</u> | true | false | ██████████ | ██████████ | ██████████ |
| <u>Exit kod:</u> | 0 | 0 | 1 | 1 | 1 |

2. Napisati funkciju `void osMkPublicDir(const char *dname);` koja pravi direktorijum 'dname' sa pravima pristupa **0777**. Testirati ovu funkciju pozivom iz *main*-a. Kao argument komandne linije program prima ime direktorijuma koji treba napraviti i poziva `osMkPublicDir` funkciju. **Nema potrebe za bilo kakvim ispisom.**

| | | | |
|------------------------------------|------------|----------------------------|------------|
| <u>Pokretanje programa:</u> | ./2 dir_a | ./2 postojeci_direktorijum | ./2 |
| <u>Standardni izlaz:</u> | ██████████ | ██████████ | ██████████ |
| <u>Exit kod:</u> | 0 | 1 | 1 |

3. Napisati funkciju `unsigned osNumOfDaysFileModified(const char *fpath);` koja vraća pre koliko je dana zadnji put sadržaj fajla promenjen u odnosu na dan kada se program pokrene (dakle **ne može se hardcode-ovati datum**). Testirati ovu funkciju pozivom iz *main*-a. Kao argument komandne linije programa prima se putanja do fajla, pozvati ovu funkciju i ispisati broj dana. **OKRENITE STRANU!**

| | | | | |
|-----------------------------|-------------------|------------------|--------------------|-------------------|
| Vreme modifikacije: | 1.txt 12.11.2016. | a.out 1.11.2016. | Nepostojeći fajl | Pogresan br. arg. |
| Pokretanje programa: | ./3 1.txt | ./3 a.out | ./3 ne_postoji.txt | ./3 |
| Standardni izlaz: | 0 | 11 | ██████████ | ██████████ |
| Exit kod: | 0 | 0 | 1 | 1 |

Test fajlove možete praviti komandom (menjajte naziv fajla, datum i vreme po potrebi):

touch 1.txt --time=mtime --date="2004-02-27 14:19:13"

4. Napisati funkciju `void osMoveFile(const char *srcPath, const char *destPath);` koja pomera fajl sa putanje `srcPath` na putanju `destPath` (kao komanda terminala `mv`). Testirati ovu funkciju pozivom iz `main-a`. Kao argumenti programa primaju se trenutna putanja fajla i putanja na koju fajl treba da se pomeri (naravno pozvati `osMoveFile` iz `main f-je`). **Nema potrebe za bilo kakvim ispisom.**

| | | | |
|-----------------------------|-----------------|-----------------|------------|
| Pokretanje programa: | ./4 1.txt 2.txt | ./4 1.txt 1.txt | ./4 1.txt |
| Standardni izlaz: | ██████████ | ██████████ | ██████████ |
| Exit kod: | 0 | 1 | 1 |

Komentar za srednji slučaj: detektovati ako su početni i rezultujući fajl isti i završiti exit kodom 1.

5. Napisati program koji sve *regularne fajlove* za koje ostali korisnici (eng. *others*) imaju `rw` prava pristupa **[1]** radi sledece:
 - a) ukoliko su modifikovani u zadnjih 30 dana **[3]** pomera ih **[4]** u odgovarajući direktorijum
 - b) ukoliko nisu modifikovani u zadnjih 30 dana **[3]** brisu se

Kao argumente komandne linije ovaj program prima putanju do direktorijuma u kome se rekurzivno traže navedeni fajlovi i naziv direktorijuma gde treba pomeriti spomenute fajlove. Potrebno je napraviti ovaj drugi direktorijum u koji se fajlovi pomeraju tako da ima prava pristupa **0777 [2]**.

Pokretanje: ./5 dirA ../backup

dirA

```
-- 1.txt -rw-rw-r-- mtime=10.10.2016.
-- 2.c  -rw-rw-rw- mtime=11.11.2016.
-- dirB
   -- OS.pdf -rw-rw-rw- mtime=5.6.2015.
   -- OS.odp -rw-rw-rw- mtime=12.11.2016.
```

Rezultat izvršavanja:

U **../backup** direktorijum su prekopirani fajlovi `2.c` i `OS.odp`. Fajl `1.txt` nema odgovarajuće privilegije. Fajl `OS.pdf` se briše zato što je vreme zadnje promene sadržaja fajla veće od 30 dana.

Napomene za ovaj zadatak:

- Pretpostaviti da se direktorijum u koji se kopira, u navedenom primeru **../backup**, NE PRAVI u okviru direktorijuma koji se obilazi rekurzivno (u navedenom primeru **dirA**).
- Ne ispisivati ništa!
- Uglaste zgrade u tekstu zadatka označavaju koji zadatak možete iskoristiti kopiranjem f-je koja je tamo implementirana.
- U slučaju da je prosleđen neispravan broj argumenata, da direktorijum koji treba obići ne postoji, kao i u slučaju drugih grešaka završiti program exit kodom 1.