

Tutustu kurssin kirjan StRa13 lukuihin 3 (File I/O) ja 4 (Files and Directories).

1. ONKO SAMA TIEDOSTO?

Kirjoita ensin funktio `int sameFile(int fd1, int fd2)`, joka saa parametrina kaksi tiedostokuvaajaa, jotka molemmat viittaavat avoimiin tiedostoihin. Funktion tulee palauttaa arvo `TRUE`, jos `fd1` sekä `fd2` viittaavat samaan fyysiseen tiedostoon, muuten arvo `FALSE`.

```
#define TRUE 1
#define FALSE 0
```

Tee sitten funktiosi testausta varten pieni pääohjelma. Millaisia erilaisia tilanteita pitää testata? Saat tehtyä tiedostolle ”alias-nimen” `ln`-komennolla, esim.

```
$ ln -s eka toka          symbolinen linkki (se suositeltava tapa)
$ ln eka koli             kova linkki
$ ls -li eka toka koli
5093699 -rw----- ... eka
5093699 -rw----- ... koli          sama indeksisolmunumero kuin eka:lla
5093742 lrwxrwxrwx ... toka -> eka   linkkitiedosto, eri indeksisolmunumero
```

Vihje: `fstat()`, tutki indeksisolmunumero ja laitenumero

2. ETSI...

Kirjoita funktio `int etsi(int fd, char *mjono)`, joka saa parametreikseen avoimen tiedostokuvaajan `fd` ja merkkijonon `mjono`. Funktio tutkii esiintyykö merkkijono jossain kohdassa tiedostoa. Jos esiintyy, funktio palauttaa arvon `TRUE`, muuten se palauttaa arvon `FALSE`.

Vihje: lue koko tiedosto kerralla riittävän suureen `malloc()`:lla varattuun merkkitaulukkoon (tiedoston koko + 1), vie taulukon loppuun merkkijonon lopetusmerkki `'\0'` ja käytä etsintään kirjastorutiinia `strstr()`. Tiedoston koon saat selville esimerkiksi `lseek()`-kutsulla. Muista sen jälkeen siirtää käsittelypositio jälleen tiedoston alkuun.

3. JA TODELLA ETSI...

Hyödynnä edellisen tehtävän funktiota. Kirjoita ohjelma, joka saa argumenttina yhden merkkijonon ja joukon tiedostonimiä. Ohjelma etsii merkkijonoa kustakin tiedostosta ja tulostaa tutkittujen tiedostojen nimet sekä löytyikö merkkijono tiedostosta.

Ohjelma käynnistetään esimerkiksi näin

```
$ a.out bingo *.txt
// komentotulkki täydentää komentoriviä ensin olemassa olevilla tiedostonimillä
// esim. a.out bingo dappu.txt puppu.txt zippu.txt
// ja käynnistää a.out-nimisen ohjelman vasta sitten
```

ja se tulostaa esimerkiksi näin

Etsittävä sana on bingo

```
dappu.txt KYLLÄ  
puppu.txt ei  
zippu.txt ei
```

Tiedostoja 3 kpl, esiintymä 1:ssä tiedostossa.

4. ONKO SAMOJA NIMIÄ

Tee ohjelma, joka vertailee kahta hakemistoa, joiden nimet annetaan komentoriviargumentteina. Ohjelma tutkii, onko hakemistoissa samannimisiä tiedostoja (tai hakemistoja). Nimet eivät ole hakemistoissa aakkosjärjestyksessä. Jokaista tiedostonimeä on siis verrattava toisen hakemiston jokaiseen tiedostonimeen.

```
$ a.out A-Dir B-Dir
```

```
Tiedosto eka löytyy molemmista hakemistoista.
```

```
Tiedosto koli löytyy molemmista hakemistoista.
```

Vihje: man 3 readdir. rewinddir()

0-0-0

Vapaaehtoinen lisäosa: tulosta myös molempien tiedostojen tyyppi. Näin aluksi riittää, että tsekkaat onko hakemisto tai tavallinen tiedosto.

```
$ a.out A-Dir B-Dir
```

```
Tiedosto eka löytyy molemmista hakemistoista.
```

```
A-Dir: tavallinen tiedosto (-)
```

```
B-Dir: tavallinen tiedosto (-)
```

```
Tiedosto koli löytyy molemmista hakemistoista
```

```
A-Dir: tavallinen tiedosto (-)
```

```
B-Dir: hakemisto (d)
```

```
Tiedosto ups löytyy molemmista hakemistoista
```

```
A-Dir: tavallinen tiedosto (-)
```

```
B-Dir: ei hakemisto tai tavallinen tiedosto
```

Meidän ympäristössä (edunix) struct dirent (ks. man 3 readdir) ei sisällä tietoa tiedoston tyypistä, joten se on tutkittava indeksisolmusta.

Vihje: man 2 stat, tiedoston tyyppi on indeksisolmun st_mode -kentässä, hyödynnä makroja S_ISREG ja S_ISDIR, myös rutiinille chdir() voi olla käyttöä.