

Döntő

2018. február 17.

Feladat: Backup alkalmazás

Sok felhasználó készít biztonsági másolatot saját adatairól. Valaki külső merevlemezt csatlakoztat a számítógépéhez, valaki NAS-t (Network Attached Storage) használ, és sokan vannak olyanok, akik valamilyen felhő alapú szolgáltatást vesznek igénybe (pl. Dropbox, Google Backup & Sync, OneDrive).

A felhő alapú backup alkalmazások általában úgy működnek, hogy az alkalmazás az elindítása után feltölti a felhőbe a kiválasztott könyvtárak fájljait, magát a könyvtárstruktúrát is, majd figyelő üzemmódba vált. Ezt követően, ha a helyi merevlemezen bármilyen változás történik, akkor az alkalmazás automatikusan feltölti a megváltozott fájlokat és a könyvtárstruktúrát a felhőbe. Így a felhasználó számára a biztonsági mentés készítése külön beavatkozás nélkül történik.

A feladatotok egy, a felhő alapú szolgáltatásokhoz hasonló backup alkalmazás készítése. Az alkalmazásnak a felhasználó számítógépén futó, ún. kliens oldali logikát kell megvalósítania.

Specifikáció

A `filerendszer-0.txt` állomány tartalmazza a számítógépen lévő összes állomány leírását. (Nem a saját számítógépeteken lévő fájlrendszert kell vizsgálnotok, hanem ez a szöveges állomány tartalmazza a feladathoz szükséges képzeletbeli fájlrendszer struktúráját.)

Példa:

```
/a/ 0 211
/a/hello.txt 123 254
/a/b/ 0 321
/a/b/read.me 100 568
```

Egy sor egy fájl vagy könyvtár leírását tartalmazza szóközzel elválasztva. Az egyes elemek jelentése:

- **Útvonal:** Az első elem a fájl vagy könyvtár teljes útvonala. Az útvonal ún. Unix formátumú, azaz előre per (/) jelekkel tagolt, pl. `/home/dusza`. (A windows-os útvonalakkal ellentétben a unixos útvonal elején nincs meghajtó, mint pl. C:.) Az útvonal mindig `/-`rel kezdődik, könyvtárak esetén `/-`rel is végződik, 128 karakternél nem hosszabb, a per (/) jelet, az ABC kisbetűit (a, b, c, ..., z), a számjegyeket (0, 1, 2, ..., 9) és a pont (.) karaktert tartalmaz. (A fájlnevekben nem lehet szóköz.) A gyökérkönyvtár mindig létezik, ezért nem szerepel az állományban.

- **Méret:** A második elem a fájl méretét jelzi bájtokban. Maximális értéke 2.000.000.000. Könyvtár esetén ez az érték mindig 0.
- **Utolsó módosítás:** A harmadik elem az az időpont (pozitív egész szám), amikor az adott fájl tartalma utoljára módosítva lett. Könyvtár esetén ez minden esetben a könyvtár létrehozásának dátuma. Maximális értéke 1.000.000, minden fájlra egyedi.

A fájlban legalább 1, és legfeljebb 256 sor szerepel. A könyvtástruktúra maximális mélysége 10, egy szinten legfeljebb 20 fájl vagy könyvtár lehet. Az állomány sorai tetszőleges sorrendben jelenhetnek meg, azaz nem feltétlenül rendezve követik egymást. Feltételezhetitek, hogy az állomány formailag helyes és egy érvényes könyvtárstruktúrát ír le. (Pl. ha egy útvonal szerepel a fájlban, akkor ezzel együtt szerepel annak összes felmenő könyvtára is.)

A `filerendszer-1.txt` fájl –amennyiben létezik–, egy későbbi állapota, „pillanatfelvétel” a számítógépen lévő összes állomány leírásának. A fájl formátuma teljesen megegyezik a `filerendszer-0.txt` fájléval.

```
/a/ 0 211
/a/b/ 0 321
/a/b/read.me 101 703
/x/ 0 600
```

Összehasonlítva a két fájlrendszert leíró állományt, megállapíthatók a következők:

- a `/a/b/read.me` fájl megváltozott (a 703-as időpillanatban, a fájl mérete 101-re változott),
- a `/a/hello.txt` fájl törlésre került (a törlés pontos időpontja nem ismert), hiszen nem szerepel már a `filerendszer-1.txt` fájlban,
- a `/x/` könyvtár létrehozásra került (a 600-as időpillanatban).

A lehetséges további `filerendszer-2.txt`, `filerendszer-3.txt` ... `filerendszer-9.txt` állományokban hasonlóan követhetőek nyomon a fájlrendszer változásai.

Fájlrendszer megtekintése

A feladatban megvalósítandó kliens oldali alkalmazásnak lehetővé kell tennie a fájlrendszer különböző állapotainak megjelenítését a képernyőn.

Egyszerre mindig egy könyvtár fájljai és alkönyvtárai láthatóak, kezdetben a gyökér könyvtaré. A felhasználó számára biztosítani kell a lehetőséget, hogy a megjelenített alkönyvtárak közül tetszőlegesen válasszon, és a kiválasztott könyvtárban lévő alkönyvtárak és fájlok jelenjenek meg. Az alkalmazásnak biztosítania kell, hogy egy szinttel feljebb lévő könyvtárban is folytatni lehessen a fájlrendszer megtekintését.

A fájlokat és könyvtárakat mindig rendezve kell megjeleníteni az aktuális könyvtáron belül. Először a könyvtárakat kell felsorolni alfanumerikus sorrendben, utána pedig a fájlok következnek szintén alfanumerikus sorrendben. Elegendő a fájlok/könyvtárak nevét és típusát (file vagy könyvtár) megjeleníteni a listázáskor.

Az elkészített programnak alkalmasnak kell lennie a fájlrendszer egyes állapotainak kiválasztására (0, 1, ..., 9 - , attól függően, hogy hány filerendszer-X.txt fájl létezik), és a különböző állapotok közötti oda-vissza léptetésre is.

Változáslista

Ahhoz, hogy a felhőbe hatékonyan kerülhessenek mentésre a változások, az alkalmazásnak képesnek kell lennie arra, hogy a fájlrendszer változásait a különböző állapotok között észrevegye. Amikor az alkalmazás a fájlrendszer egy adott állapotát megjeleníti, jeleltesse meg ezzel egyidejűleg listaszerűen a fájlrendszer előző állapota óta bekövetkezett változásokat is!

A 0. állapothoz tartozó változáslista üres, mert ez a kiinduló állapota a fájlrendszernek.

Az 1-es állapothoz tartozó változáslista a következő:

```
LETREHOZ /x/ 0 600
MODOSIT /a/b/read.me 101 703
TOROL /a/hello.txt
```

- A LETREHOZ-zal kezdődő sorok az újonnan született fájlokat vagy könyvtárakat mutatják. A sor tartalmazza a fájl vagy könyvtár teljes útvonalát, annak méretét és a változás bekövetkeztének időpontját.
- A MODOSIT-tal kezdődő sorok azokat a fájlokat írják le, amelyek a fájlrendszeren mindkét időpillanatban léteztek, de változásra kerültek. A sor tartalmazza a fájl vagy könyvtár teljes útvonalát, annak méretét és a változás bekövetkeztének időpontját.
- A TOROL-lel kezdődő sorok adják meg a törölt fájlokat vagy könyvtárakat. A sor tartalmazza a fájl vagy könyvtár teljes útvonalát.

A sorok tetszőleges sorrendben megjeleníthetők.

Speciális fájlok kezelése

Vannak olyan fájlok, amelyekről az alkalmazás nem készít mentést, ezért a változáslistának a következő fájlokat nem szabad tartalmaznia:

- Rendszerfájlok: amelyeknek a neve `.(pont)`-tal kezdődik (pl. `/a/b/.haho`)
- Temp fájlok: amelyeknek a kiterjesztése `.tmp`
- Nagy méretű fájlok: amelyeknek a mérete 100MB feletti ($100 \cdot 2^{20}$ bájt).

Rekurzív törlés

Amennyiben egy könyvtárban az összes alkönyvtár és az azokban szereplő fájlok rekurzív (beleértve magát a kiinduló könyvtárat is) törlésre kerülnek, akkor a változáslistába elegendő a következő formátumú sort kiírni:

```
REKURZIV-TOROL /x/
```

Beadandó

- A program forráskódja és a lefordított/futtatható állomány (Az állomány neve: backup)
- A fejlesztői dokumentáció: Szöveges (elektronikus) dokumentum, amelyben ismertetitek a legfontosabb elnevezéseket (változók, osztályok, eljárások, függvények neve), és az egyes részfeladatokban alkalmazott módszereket (az algoritmus működési elvét).

A bemutatáshoz külön szemléltető anyag készítése (pl. PowerPoint prezentáció) nem kötelező, de ha készül ilyen, azt is be kell adni!

A munka szóbeli bemutatása:

Szemponatok, ajánlott vázlat:

- A feladat előkészítésének bemutatása, a feladatok szétosztásának elvei
- Az elkészített program bemutatása
 - A felhasználó számára
 - A program működésének ismertetése bemutatással
 - A fejlesztő számára
 - A program szerkezetének ismertetése

Minden csapattagnak részt kell vennie a munka bemutatásában!

Elérhető pontszám: 150 pont.

Jó munkát kíván a Versenybizottság!