

JEGYZŐKÖNYV

Számítógép Architektúrák

Féléves feladat

1.Feladat

Készítette: **Ferencsik Róbert**

Neptunkód: **BQLOTW**

Dátum: 2023.11.30

Tartalomjegyzék

1. Bevezetés.....	3
1.1. Dokumentum célja.....	3
1.2. Saját gondolatok.....	3
2. Követelmény elemzés.....	3
2.1. A feladat kiírás és értelmezése.....	3
2.2. Döntés a tartalomról.....	3
3. A szkript elemei és feladatuk.....	4
3.1. Mappa ellenőrzése elkészítése.....	4
3.2. Zip törlése.....	4
3.3. Fájlok törlése.....	4
3.4. Letöltés, kibontás, és ezek sikeressége... ..	5
3.5. Szűrés.....	5
4. Fejlesztés és tesztelés.....	5

Bevezetés

A számítógép Architektúrák második beadandó feladatának dokumentációját tartja kezében. A feladat bash script elkészítése volt. A program működéséről írtam részletesebben, és az elkészítés folyamatáról is szó lesz alább.

Mivel betegség miatt egyedül készültem fel így kihívásnak bizonyult a feladat, azonban türelemmel és próbálkozások sorával elkészítettem az első saját szkriptemet. Nagyon izgalmas és motiváló gondolat, hogy a saját eszközömmön az unalmas egyszerű, és sokszor elvégzendő feladatokat automatizálhatom, ezzel kiváltva őket, egy egyszeri összetettebb programozási feladatra, majd annak az elkészítése után egy egyszerű futtatásra.

Követelmény elemzés

A feladat kiírása alapján az MVK Zrt GTFS adatbázisát kellett a szkriptnek letöltenie, amely egy zip fájl. Szóval kitömörítés után rá kellett szűrnie a letöltött adatbázisok egyikében azokra az útvonalakra mely a saját lakóhelyhez legközelebbi megállóból indul, vagy odaérkezik.

Mind ezek mellett ellenőrzéseket is kellett végeznie, még hozzá sorban a következőket. Ha a letöltendő fájl már egyszer le volt töltve, akkor az újbóli letöltés előtt törölje az előző verziót! Ha egy mappába már korábban ki lett tömörítve a letöltött állomány, akkor az újbóli kitömörítés törölje a mappa tartalmát!

2. feladat – Mentés: *lakohely_gtfs.sh*

Az MVK Zrt. elérhetővé tesz egy szabványos GTFS adatbázist a fejlesztők számára, hogy a *menetrendi adatokat* a saját applikációba tudják integrálni.

Írjon egy shell script fájlt, amely *letölti ezt az adatbázist, és kilistázza belőle azokat az utakat, amelyek a Lakóhelyből indulnak, ill. a Lakóhelybe érkeznek!*

Parancsok: *wget, unzip (kitömörítésre), cat, grep*

Szükséges ellenőrzések!

Ha a letöltendő fájl már egyszer le volt töltve, akkor az újbóli letöltés előtt törölje az előző verziót!

Ha egy mappába már korábban ki lett tömörítve a letöltött állomány, akkor az újbóli kitömörítés törölje a mappa tartalmát!

Ha nincs ilyen menetrendi adatok, akkor Miskolc várost használják a feladat elkészítéséhez!

Az eredeti feladat kiírás szövege

A szkriptet *lakohely_gtfs.sh* néven kell menteni, és beadáskor látszódnia kell a lefutott program eredményének, amit én úgy oldottam meg, hogy *.txt* fájlba írja ki a szűrés eredményét a programom. Így látszódik, hogy sikeresen megtörtént a szűrés, de igazából az elérési útvonal változóban van tárolva, úgyhogy annak a módosításával lehet használni másik eszközön is.

A szkript elemei és feladatuk

Ahogy már írtam a jegyzék útvonala változóban van tárolva *output_folder* néven, így könnyen kezelhető a programon belül és a letöltendő fájl url címe is változóban van tárolva, így letisztultabb és könnyen változtatható a program.

Az első lépésben ellenőrzésre kerül egy *if* feltételében, hogy a jegyzék amiben szeretnénk “dolgozni” létezik-e. Maga a feltétel pontosan így szerepel a kódban: *[! -d "\$output_folder"]* ahol a felkiáltójel negálja az értelmét annak, hogy a dollár jel után lévő változóban tárolt nevű fájl létezik, és jegyzék amit a *-d* vel adunk meg a programnak. Természetesen ha az előbbi feltétel igazat ad vissza, az azt jelenti, hogy nincs ilyen nevű jegyzék, és így belépve a feltételbe egy *mkdir* paranccsal elkészít azt.

Majd a *cd* megváltoztatja az aktuális munka könyvtárat a GTFS nevű vagy létező vagy létrehozott jegyzékre. A *//* vagyfeltétel után lévő *exit* pedig kilép a végrehajtásból, ha nem sikerült a könyvtár váltás.

A következő két feltételes elágazás sorban a következőkért felelős: Vizsgálat, hogy a letölteni kívánt tömörített fájl létezik-e (*-e* avagy *exists* szűréssel), amennyiben létezik, kiírja a konzolra, hogy mit csinál épp, és letörli a fájlt magát. Majd ez megtörténik a mappa tartalmára is, azt nézi

a program feltétele, hogy létezik-e elem a munkakönyvtárban (`["$(ls -A .)"]` nem veszi figyelembe a speciális `..` fájlokat). Ha a kiértékelés igaz, akkor `rm -r *` al rekurzívan törli a munkakönyvtárban lévő összes fájlt és alkönyvtárat.

```
if [ -e "downloaded_file.zip" ]; then
    echo "Deleting previous version of downloaded ZIP file."
    rm downloaded_file.zip
fi

if [ "$(ls -A .)" ]; then
    echo "Deleting previous content of the extraction folder."
    rm -r *
fi
```

A fent említett két feltételes elágazás

A letöltés után egymásba ágyazva két feltétel következik melyek csak akkor futnak le ha az előttük lévő utasítás vissza térési értéke 0 melyet a program kódban így tudunk megadni feltételként `[$? -eq 0]`. Szóval ha a letöltés sikeres volt akkor a *unzip* utasítással kicsomagoljuk a letöltött fájlt, és ha az is sikeres akkor a sikeresség tényét kiíratjuk a képernyőre ellenkező esetekben *exit 1* et adunk vissza. Itt az egyes jelzi, hogy hiba történt futás közben.

Majd létrehozuk a *lakohelyutvonal.txt* fájlt amit eltárolunk egy változóba rögtön utána, és rászűrünk a megállóral melyre szeretnénk és adjuk is tovább a fájlunkba a szűrés eredményét egy `>>` csővezeték operátorral. Több fájlt is tartalmaz kicsomagolás után a mappa, de a *routes.txt* re volt szükségem, a végén egy egyszerű ellenőrzés, ugyan úgy az előző lépés sikerességét nézve kiírja, hogy sikeres volt a szűrés, vagy sem.

Fejlesztés és tesztelés

Koránt sem tekintem véglegesnek a programom, azonban a sok teendőm mellett próbáltam annyira egyszerűen megoldani a feladatot amennyire lehet. Ez magában a kódban nyilvánul meg, hiszen egy egyszerű “spagetti” kódot készítettem. Szeretném majd a jövőben átdolgozni, szépen kiszedni függvényekbe a funkciókat, és másfajta megoldásokat is esetleg tanulmányozni, készíteni. A feladat elkészítéséhez találtam egy hasznos weboldalt, amin Alexandra Dobosné munkája tekinthető meg egy prezentáció formájában. Az alapján szépen sorban oldottam meg a kis részekre felosztott feladatot.

Számítógépes alapismeretek Shell script programozás - ppt letölteni (slideplayer.hu)

A részek amiken egymás után külön dolgoztam megegyeznek az előző részben ahogyan tagoltam a program működését. Nem találtam annyira nagy feladatnak, hogy előre készítsek

róla egy tervet, habár érdemes lett volna, hogy azt is gyakoroljam (megint csak az idő gazdálkodásom tudom felhozni mentségként).

Maga a tesztelés is elég egyenesen ment. Megírtam egy részt, lefuttattam többször, úgy hogy a feltételek mindkét ága előjőjön és ha nem az történt amit szerettem volna, akkor olvasás elírás keresése míg meg nem oldottam.