

## 1A Házi feladat

### Címtárak

**FIGYELEM:** A házi feladat megoldása előtt olvassa el a tárgy weblapján lévő HF tudnivalókat! A házi feladat leadása előtt nézze végig a HF tudnivalóknál szereplő ellenőrző listát!

### **Feladat**

A *SuperHotels* szállodalánc felvásárolta az egyik konkurenciáját, és szeretnék az informatikai rendszereket egységesíteni. Mindkét szervezet eredetileg saját címtárban tárolta a felhasználói adatokat, és ezek összefűzését kell megoldani az integrációhoz. A rendszerek üzemeltetéséért felelős informatikusok egy jól átgondolt lépéssel a felvásárolt konkurencia címtárát egyszerűen importálták a szállodalánc központi címtárába és itt kezdődtek a problémák.

Amint az utólag kiderült, bizonyos Linux rendszerek ezen adatbázisok alapján hitelesítik a felhasználóikat, de sajnos felhasználó-ütközések keletkeztek, mert a régi és az importált címtárakban megegyező UID-del rendelkező felhasználók voltak.

Készítsünk olyan szkriptet, amely elősegíti a hiba elhárítását! A szkript két megadott szervezeti egységben (*Organizational Unit*) tetszőlegesen mély hierarchiaszinten keresse meg az összes olyan alkalmazottat, akinek megegyezik a felhasználói azonosítója (*uid* adata).

Az LDAP struktúrában a `dc=irf,dc=local` csomópont a gyökérelem és a következő felhasználó jogosult itt keresni:

- Felhasználó: `cn=root,dc=irf,dc=local`
- Jelszó: `LaborImage`
- LDAP szerver: `localhost`

(Az LDAP elérést végző felhasználó rögzítése a szkriptben csak a házi feladat könnyítése miatt szerepel, éles környezetben ez nem javasolt, ott például Kerberos hitelesítés használata ajánlott.)

A kiadott virtuális gépben található adatbázisban ezek a hibák nem találhatók meg, a szkript teszteléséhez azokat kézzel kell elhelyezni benne.

### **A szkript elnevezése és paraméterezése**

```
list-duplicated-uid.sh -d <old_ou> -n <new_ou> -o <output>
```

A szkriptnek kötelező ezt az elnevezést és paraméterezést használnia!

A szkript a paramétereket a következő formában fogadja:

- `old_ou`: az eredeti szervezeti egység DN-je, amiben keresünk (kötelező megadni)
- `new_ou`: a felvásárolt szervezeti egység DN-je, amiben keresünk (kötelező megadni)

- output: a kimeneti CSV fájl neve (kötelező megadni)

Álljon itt egy példa a szkript egy lehetséges, helyes használatára:

```
list-duplicated-uid.sh -d "ou=France,ou=Europe,ou=Hotels,dc=irf,dc=local"
-n "ou=Italy,ou=Europe,ou=Hotels,dc=irf,dc=local" -o output.csv
```

### Kimeneti fájl

A kimeneti fájl egy CSV fájl a következő oszlopnevekkel: DN1, DN2, UID, ahol a DN1 az eredeti alkalmazott DN-je, a DN2 pedig a felvásárolt szervezet alkalmazottjának DN-je és az UID pedig az ütköző, közös azonosítójuk.

Példa:

DN1, DN2, UID

```
"cn=cvanness,ou=Lyon,ou=France,ou=Europe,ou=Hotels,dc=irf,dc=local", →
"cn=aevelyn,ou=Osaka,ou=Japan,ou=Asia,ou=Hotels,dc=irf,dc=local", →
"3491"
"cn=ahyde,ou=Paris,ou=France,ou=Europe,ou=Hotels,dc=irf,dc=local", →
"cn=dgarnet,ou=Osaka,ou=Japan,ou=Asia,ou=Hotels,dc=irf,dc=local", →
"3471"
```

A fejlécben szereplő oszlopneveknek és az oszlopok sorrendjének pontosan a mintát kell követnie. A felesleges " jelek elhagyhatók. A sorok sorrendje tetszőleges. A → jelek természetesen nem szerepelnek a kimeneti fájlban, csak azt jelölik, hogy a tényleges kimeneti fájlban a jelzett helyen nincsen sortörés (a fenti minta három sorból áll).

### További elvárások

- Figyeljen a hibakezelésre, ellenőrizze, hogy a megadott szervezeti egységek léteznek-e! Ellenőrizze továbbá, hogy a megadott kimeneti fájl nem létezik! Szintén legyen hibajelzés akkor, ha egy kötelező paraméter nem került megadásra.
- Igyekezzen elkerülni a felesleges címtárlekérdezéseket! A nagy költségű szűréseket is a szerveren végezzük el!