

Budapesti Műszaki és Gazdaságtudományi Egyetem  
Intelligens rendszerfelügyelet (VIMIA370)

## **2C Házi feladat**

Konfigurációkezelés

Bori András (IA10CQ)

2011. április 9.

## 1 Bevezető

### 1.1 A házi feladat célja

Elveszünk mindig a sok hálózati beállítás és topológia ábra részleteiben, ezért kéne egy gyors áttekintő kép a gépeink fő hálózati beállításairól. Készítsen tehát egy olyan Bash scriptet, ami paraméterként megkapja egy CSV fájl elérési útját, majd a CSV fájlban felsorolt gépeken futó CIMOM-okat sorban lekérdezi CIM-XML segítségével, és mindegyikről kigyűjti a következő információkat. Fel kell sorolni, hogy milyen hálózati interfészek vannak beállítva rajtuk, és azok milyen fizikai címmel rendelkeznek, majd minden hálózati interfészhez meg kell jeleníteni a hozzá tartozó IP paramétereket (IP cím, alhálózati maszk). Az eredményt egy egyszerű szöveges jelentésben prezentálja, ahol minden géphez tartozik egy interfész táblázat. Ennek sorai tartalmazzák az interfész nevét és fizikai címét, valamint IP címenként egy-egy sorban az IP címet és alhálózati maszkot.

#### A szkript elnevezése és paraméterezése:

```
getInterfaces.sh <machineData> <outputFile>
```

- <machineData>: a távoli számítógépek adatait tartalmazó CSV fájl, kötelező.
- <outputFile>: a kimeneti fájl elérési útja, kötelező. [2]

### 1.2 Felhasznált környezet

- CentOS VMware virtuális gép [3]
- ECUTE Explorer [4]

Két CentOS virtuális gépet üzemeltem be. Az egyik géphez hozzáadtam további két hálózati interfészt, valamint megváltoztattam a gép nevét irf2011-ről bme2011-re, hogy meg tudjam különböztetni őket.

A virtuális gépeken az alábbi beállításokat végeztem el, hogy a CIM szerver működjön és elérhető legyen:

- **CIM szerverhez a hozzáférés beállítása:**

```
nano /etc/Pegasus/access.conf
```

- **CIM kiszolgáló elindítása:**

```
sudo /etc/init.d/tog-pegasus start
```

- **Milyen portokon figyel a gép:**

```
sudo netstat -t -l -p --numeric
```

A CIM server a következő portokon érhető el: 5988 (http) és 5989 (https)

## 2 Megoldás

A megoldás során öt fontos műveletsort végeztem el, amelyeket az alábbiakban részletesen kifejték.

## 2.1 Bemeneti paraméterek vizsgálata

A script a specifikációnak megfelelően két paramétert vár, a bemeneti és a kimeneti fájl elérési útját. Ezek közül mindkettő megadása kötelező, azaz ha nem adjuk meg őket, akkor a script hibaüzenetet ír ki és leáll.

```
if [ $# -ne 2 ]; then
    echo 'A bemeno parameter egy input es egy output fajlnev kell legyen!';
```

További elvárás, hogy a bemeneti igen, a kimeneti fájl viszont ne létezzen (hogy ne írjunk felül létező fájlt).

```
elif [ ! -f $1 ]; then
    echo 'A bemeneti fajl nem letezik!';
elif [ -f $2 ]; then
    echo 'A kimeneti fajl mar letezik!';
```

Továbbá szükséges, hogy amennyiben a bemeneti fájl létezik, az tartalmazzon valamit, egyébként nem tudunk mit kezdeni vele.

```
if [ -z "$machineData" ]; then
    echo "Hiba: A(z) '${1}' fajl ures."
```

## 2.2 Adatok beolvasása a bemeneti fájlból

A bemeneti CSV fájlban (*machineData*) minden sorban öt adat van (gép címe, port, protokoll, felhasználónév és jelszó), ezek vesszővel vannak elválasztva. Mindegyik adat számára létrehoztam egy tömböt és a fájl tartalmát soronként beolvasva beleteszem őket.

```
i=0;
for row in $machineData; do
    machineName[$i]=`echo $row | cut -d ',' -f 1`;
    port[$i]=`echo $row | cut -d ',' -f 2`;
    protocol[$i]=`echo $row | cut -d ',' -f 3`;
    user[$i]=`echo $row | cut -d ',' -f 4`;
    password[$i]=`echo $row | cut -d ',' -f 5`;

    i=$((i+1));
done;
```

A beolvasás után leellenőrzöm, hogy a fájl formátuma megfelelő-e, azaz megvizsgálom az első sorát, ahol 'machineName,port,protocol,user,password'-nek kell szerepelnie (pontosabban a beolvasott tömbök nulladik elemét vizsgálom meg, ahol sorra ezeknek kell szerepelni). Ha ez nem teljesül, akkor a bemeneti fájl valószínűleg nem megfelelő, ezért a script hibaüzenetet ír ki és leáll.

```
if [ ${machineName[0]} != 'machineName' ] || [ ${port[0]} != 'port' ] ||
    [ ${protocol[0]} != 'protocol' ] || [ ${user[0]} != 'user' ] ||
    [ ${password[0]} != 'password' ];
then
    echo "Hiba: A(z) '${1}' fajl formatuma nem megfelelo."
    exit 5;
fi
```

### 2.3 Hálózati interfészekre és IP címekre vonatkozó adatok lekérdezése

Az Ecute Explorer segítségével megkerestem azt a két CIM osztályt, amelynek az példányai tartalmazzák a hálózati interfészekre és IP címekre vonatkozó információkat. Ezek a CIM\_NetworkPort és a CIM\_IPProtocolEndpoint.

Az adatok lekérésére írtam egy felparaméterezett lekérdezést, ahol a paraméterek helyére iterációnként az adott gépre vonatkozó elérhetőségi adatokat írom.

A lekérdezett adatok közül egy szűréssel csak a megfelelő attribútumokra vonatkozó sorokat tartom meg, ez a hálózati interfész esetén az interfész neve és MAC címe, IP esetén az interfész neve, az IP cím (csak IPv4-es IP címet kérdeztem le, mert jelenleg még ez az elterjedtebb) és az alhálózati maszk. Továbbá egy vágással a kapott sorokból csak az attribútum értékét tartom meg (ami idézőjelek között van).

```
ethernet=`wbemcli -nl ei
    "${protocol[$i]}://${user[$i]}:${password[$i]}@${machineName[$i]}:${port[$i]}
    /root/cimv2:CIM_NetworkPort"
    2>/dev/null | egrep "^-(ElementName|PermanentAddress)" | cut -d '"' -f 2`

IPinfo=`wbemcli -nl ei
    "${protocol[$i]}://${user[$i]}:${password[$i]}@${machineName[$i]}:${port[$i]}
    /root/cimv2:CIM_IPProtocolEndpoint"
    2>/dev/null | egrep "^-(ElementName|IPv4Address|SubnetMask)" | cut -d '"' -f 2`
```

Ezek után a lekérdezett adatokat tömbökbe (*eth* és *ip*) teszem, hogy könnyebben kezelhetőek legyenek.

```
for line in ${ethernet}; do
    eth[$j]=`echo $line`
    j=$((j+1));
done;

for line in ${IPinfo}; do
    ip[$j]=`echo $line`
    j=$((j+1));
done;
```

### 2.4 Lekérdezett adatok kiírása a kimeneti fájlba

Egy *for* ciklussal végigmegegyek a beolvasott hálózati interfészek tömbjén. Amennyiben az adott interfész a *loopback* hálózati interfész ("*lo*" a neve és nincs MAC címe, mert egy virtuális hálózati interfész), akkor továbblépek a következő elemre (ezt nem találtam fontosnak kiírni a kimenetbe).

```
if [ "${eth[$j]}" = lo ]; then continue; fi
```

Kiírom a kimeneti fájlba az adott interfész nevét és címét, majd végigmegegyek az IP-re vonatkozó adatokat tartalmazó tömbön.

```
echo "-- ${eth[$j]}, ${eth[$j+1]}" >> $outputFile
for ((k=0;k<${#ip[@]};k++)); do
```

Megvizsgálom, hogy az adott interfészhez tartoznak-e az IP adatok ("IPv4\_interfész neve" formátumban volt megadva a lekérdezett IP címhez tartozó interfész neve), ha igen, akkor kiírom a következő IP címet és hálózati maszkot a kimeneti fájlba.

```
temp=`echo ${ip[$k]} | cut -d '_' -f 1`
if [ "$temp" = "IPv4" ]; then
    interface=`echo ${ip[$k]} | cut -d '_' -f 2`
    if [ "$interface" = "${eth[$j]}" ]; then write=1
    else write=0;
```

```
    fi
    k=$((k+1));
fi
if [ $write -eq 1 ]; then
    echo "    ${ip[$k]},${ip[$k+1]}" >> $outputFile
    k=$((k+1));
fi;
```

## 2.5 Lekérdezés eredményességének vizsgálata

A szkript végén megvizsgálom, hogy létezik-e a kimeneti fájl. Amennyiben nem létezik, kiírok egy hibaüzenetet, hogy a felhasználó tudja, hogy miért nem jött létre a kimeneti fájl. Ez az eset akkor történhet meg, ha a bemeneti CSV fájlban nem volt megadva adat a fejlécen kívül, vagy a megadott gépek nem voltak elérhetőek.

```
if [ ! -f $outputFile ]; then
    echo 'Hiba: Nincs a megadott parameterek alapján lekerdezhető információ.'
fi
```

## 3 Tesztelés

A tesztelés során kitértem az eredményes és az eredménytelen működésre.

### 3.1 Tesztelés: eredménytelenül

#### 3.1.1 Paraméterek megadása nélkül

A scriptet paraméterek megadása nélkül futtattam, hogy megnézzem megfelel-e annak a követelménynek, hogy bizonyos paraméterek megadása kötelező.

- **Utasítás:**

```
./getInterfaces.sh
```

- **Hibaüzenet:**

Hiba: Ket bemeneti parametert kell megadni, egy bemeneti es egy kimeneti fajl eleresi utjat.

- **Értékelés:**

Mivel két paraméter megadása kötelező, ezért ezek hiánya miatt hibaüzenetet kapunk.

#### 3.1.2 Kimeneti fájl megadása nélkül

A paraméterezésben megadtam a bemeneti fájlt, viszont a kimeneti fájlt nem, hogy látni lehessen, hogy annak a megadása is kötelező.

- **Utasítás:**

```
./getInterfaces.sh input.csv
```

- **Hibaüzenet:**

Hiba: Ket bemeneti parametert kell megadni, egy bemeneti es egy kimeneti fajl eleresi utjat.

- **Értékelés:**

Mivel pontosan két paramétert vár a script, ezért az eredmény az előzőhöz hasonló.

### 3.1.3 Nem létező bemeneti fájl

Egy még nem létrehozott bemeneti, illetve kimeneti fájlt adtam meg a paraméterezésben.

- **Utasítás:**

```
./getInterfaces.sh aaa.csv output.txt
```

- **Hibaüzenet:**

Hiba: A(z) 'aaa.csv' fájl nem letezik.

- **Értékelés:**

A számítógépek adatait tartalmazó 'abc.csv' állomány nem létezik, tehát nem áll rendelkezésre olyan adat, ami alapján lekérdezést lehetne indítani.

### 3.1.4 Üres bemeneti fájl

Egy létező, viszont üres bemeneti fájlt adtam meg.

- **Utasítás:**

```
./getInterfaces.sh input1.csv output.txt
```

- **Hibaüzenet:**

Hiba: A(z) 'input1.csv' fájl ures.

- **Értékelés:**

Az előző teszthez hasonlóan nem áll rendelkezésre olyan adat, amivel a program tovább futhatna.

### 3.1.5 Rossz formátumú bemeneti fájl

Egy létező, viszont nem megfelelő formátumú (nincs megfelelő fejléce) bemeneti fájlt adtam meg.

- **input2.txt**

```
localhost,5988,http,meres,LaborImage
```

- **Utasítás:**

```
./getInterfaces.sh input2.csv output.txt
```

- **Hibaüzenet:**

Hiba: A(z) 'input2.csv' fájl formatuma nem megfelelo.

- **Értékelés:**

A bemeneti fájl első sorában a fájl soraiban található adatok megnevezését kell felsorolni, ami a specifikáció szerint: machineName,port,protocol,user,password

Mivel ez nem teljesül, ezért nem biztos, hogy a bemeneti állomány (megfelelő sorrendben) megfelelő adatokat tartalmaz, ezért nem lehet vele dolgozni.

### 3.1.6 Nem elérhető számítógép és nem található eredmény

A bemeneti állományban egy nem létező gép adatait adtam meg.

- **input3.csv:**

```
machineName,port,protocol,user,password  
123.123.123.123,1234,http,felhasznalo,jelszo
```

- **Utasítás:**

```
./getInterfaces.sh input3.csv output.txt
```

- **Hibaüzenet:**

```
Figyelmeztetés: http://felhasznalo@123.123.123.123:1234 nem erhető el.  
Hiba: Nincs a megadott paraméterek alapján lekérdezheto információ.
```

- **Értékelés:**

Először egy figyelmeztetést kaptunk arról, hogy a megadott számítógép nem érhető el. Majd, mivel nincs több gép megadva a bemeneti fájlban, ezért a szkript jelzi, hogy nem sikerült semmilyen információt lekérdeznie.

### 3.1.7 Már létező kimeneti fájl

Ugyanazt a létező állományt adtam meg bemeneti fájlként, mint kimeneti fájlként (az előző tesztben használt input3.csv fájlt).

- **Utasítás:**

```
./getInterfaces.sh input3.csv input3.csv
```

- **Hibaüzenet:**

```
Hiba: A(z) input3.csv fajl mar letezik.
```

- **Értékelés:**

Mivel nem szabad megengedni, hogy a program egy létező állományt önkényesen felülírjasson, ezért a létező 'input3.csv' fájlt nem módosíthatja, így a script futása leáll.

## 3.2 Tesztelés: eredményesen

### 3.2.1 A helyi és egy távoli gép lekérdezése

Két létező számítógép adatait adtam meg a bemeneti fájlban.

- **input01.csv:**

```
machineName,port,protocol,user,password  
localhost,5988,http,meres,LaborImage  
192.168.0.3,5989,https,pegasus,LaborImage
```

- **Utasítás:**

```
./getInterfaces.sh input01.csv output01.txt
```

- **output01.txt:**

```
++ irf2011.localdomain
-- eth0, 00:0C:29:D7:B3:AB
   192.168.198.130,255.255.255.0

++ bme2011
-- eth0, 00:0C:29:6E:05:DB
   192.168.198.131,255.255.255.0
-- eth1, 00:0C:29:6E:05:E5
   192.168.0.3,255.255.255.0
-- eth2, 00:0C:29:6E:05:D1
   192.168.198.132,255.255.255.0
```

- **Értékelés:**

A kimenet a specifikációnak megfelelő és a lekérdezett adatok helyesek.

### 3.2.2 A helyi és egy távoli gép lekérdezése + nem elérhető gépek

Az előző teszt bemenetét kiegészítettem hibás adatokkal. Először rossz jelszót, aztán rossz protokollt majd rossz felhasználónevet adtam meg az egyébként létező és elérhető gépekhez.

- **input02.csv:**

```
machineName,port,protocol,user,password
192.168.0.3,5989,https,pegasus,rosszjelszo
localhost,5988,http,meres,LaborImage
localhost,5988,https,meres,LaborImage
192.168.0.3,5989,https,pegasus,LaborImage
localhost,5989,https,nemletezo,jelszo
```

- **Utasítás:**

```
./getInterfaces.sh input02.csv output02.txt
```

- **Hibaüzenet:**

```
Figyelmeztetes: https://pegasus@192.168.0.3:5989 nem erhető el.
Figyelmeztetes: https://meres@localhost:5988 nem erhető el.
Figyelmeztetes: https://nemletezo@localhost:5989 nem erhető el.
```

- **output02.txt:**

```
++ irf2011.localdomain
-- eth0, 00:0C:29:D7:B3:AB
   192.168.198.130,255.255.255.0

++ bme2011
-- eth0, 00:0C:29:6E:05:DB
   192.168.198.131,255.255.255.0
-- eth1, 00:0C:29:6E:05:E5
   192.168.0.3,255.255.255.0
-- eth2, 00:0C:29:6E:05:D1
   192.168.198.132,255.255.255.0
```

- **Értékelés:**

A kimenet az előbbi teszt kimenetével megegyezik. A hibásan megadott gépek ellenére nem áll le a program futása, hanem figyelmeztetést ír ki.

### Hivatkozások

- [1] BME MIT, *Hogyan készítsünk házi feladat dokumentációt és mérési jegyzőkönyvet*, elérhető online: <http://www.inf.mit.bme.hu/edu/dokumentacio>
- [2] BME MIT Intelligens rendszerfelügyelet, *Házi feladat kiírások / 2-C Konfigurációkezelés / 2C-IRF-HF-v2.pdf*: <https://sauron.inf.mit.bme.hu/Edu/IRF/irf2011.nsf>
- [3] BME MIT, *CentOS 5.5*: <http://static.inf.mit.bme.hu/edu/irf/virtualmachines/CentOS-5.5-IRF-v2.7z>
- [4] *ECUTE Explorer*, <http://sourceforge.net/apps/mediawiki/sblim/index.php?title=Ecute>