

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav telekomunikácií

Riadenie telekomunikačných systémov

SNMP

Dátum: 20.10.2011
Ak. rok: 2011/2012

Bc. Róbert Grenčík

Zadanie

Zadanie číslo 66.

Napíšte program, ktorý:

1. zo súboru načíta názov alebo adresu zariadenia a názov komunity
2. načíta a vypíše na obrazovke počítača tabuľku hrPrinterTable zadaného zariadenia.

hrPrinterTable, MIBHOST-RESOURCES-MIB, {1.3.6.1.2.1.25.3.5}

Riešenie

Na začiatku som si vytvoril súbor s názvom `config.txt`, v ktorom je v prvom riadku uvedená IP adresa zariadenia, v druhom názov komunity a v treťom je uvedené OID tabuľky, ktorú potrebujem vypísať.

```
147.175.104.190
public
iso.3.6.1.2.1.25.3.5
```

Následne som vytvoril zdrojový kód, potrebný pre vypísanie danej tabuľky. Zdrojový kód je uvedený nižšie s príslušným popisom činností.

Najprv som si načítal premenné zo súboru do premennej `config` typu pole. To som dosiahol načítaním jednotlivých riadkov zo súboru cez cyklus `while`.

```
#!/bin/bash
declare -a config
let count=0

while read line
do
    config[$count]=$line
    ((count++))
done<config.txt
```

Ďalej som vypísal hlavičku tabuľky a zadefinoval som si premenné, ktoré neskôr využijem. V premennej `vyrezane` sa ukladá vrátený OID z prvého volania `snmpgetnext`. V premennej `tabla` sa ukladá OID, ktoré slúži na vybratie hodnoty v `snmpget`, ktoré sa neskôr aj vypíše. Do premennej `riadok` je vložená hodnota 1, ktorá sa v ďalšom inkrementuje a vyjadruje nám číslo riadka tabuľky. Parameter `e` pri príkaze `echo` mi zaručí to, že sa bude vypisovať `\t` ako tabulátor.

```
echo "Tabulka hrPrinterTable"
echo -e "Riadok\t\tStav\t\tChyba"
```

```
vyrezane=${config[2]}
tabla=${config[2]}
riadok=1
```

Po zadefinovaní premenných som vytvoril cyklus typu `while`, v ktorom sa vytvárajú jednotlivé riadky tabuľky. Tento cyklus sa bude vykonávať kým sa bude rovnať premenná `vyrezane` tretej hodnote poľa `config`, čo je vlastne OID tabuľky. Na konci zdrojového kódu je ukončenie cyklu `while` a taktiež inkrementácia premennej `riadok`.

```
while [[ $vyrezane = ${config[2]} ]]
do
```

Keďže jednotlivé riadky mám zabezpečené cyklom `while`, potrebujem zabezpečiť vypisovanie jednotlivých stĺpcov. Stĺpce som zabezpečil cyklom typu `for` pre hodnoty 0, 1 a 2, nakoľko z RFC viem, koľko stĺpcov mám očakávať. Na začiatku cyklu si pomocou `snmpgetnext` vyberiem hodnotu podľa OID uloženého v premennej `tabla`, následne sa z neho vyreže vrátené

OID.

```
for cislo in 0 1 2
do
row=$(snmpgetnext -v1 -c ${config[1]} ${config[0]} $tabla)
vyrezane=$(echo $row | cut -c -${#config[2]})
```

Na začiatku cyklu sa overuje podmienka, že či sa už nedostal výber riadkov do ďalšej tabuľky. Ak áno, tak sa volá `break 2`, ktorý ukončuje cyklus `for` a `while`, čo znamená ukončenie celého programu.

Následne sa porovnáva číslo stĺpca a podľa toho sa prispôsobuje výpis. Pre nultý stĺpec sa vypíše len poradové číslo riadka. V prvom a druhom stĺpci sa vybratá hodnota zamieňa za slovné znenie hodnoty. Na začiatku každého vypisovania stĺpca sa vyberá hodnota podľa aktuálne nastaveného OID v premennej `tabla` pomocou príkazu `snmpget`.

```
if [[ $vyrezane = ${config[2]} ]]; then
    if [ $cislo -eq 0 ]; then
        echo -n -e "$riadok\t\t"
    elif [ $cislo -eq 1 ]; then
        tabla=${config[2]}.1.$cislo.$riadok
        posledny=$(snmpget -v1 -c ${config[1]} ${config[0]}
$tabla)
        hodnota=$(echo $posledny | cut -d ' ' -f4)

        case "$hodnota" in
            1) vypis="Other" ;;
            2) vypis="Unknown" ;;
            3) vypis="Idle" ;;
            4) vypis="Printing" ;;
            5) vypis="Warmup" ;;
            *) vypis="Neznama hodnota" ;;
        esac

        echo -e -n "$vypis\t\t"
    elif [ $cislo -eq 2 ]; then
        tabla=${config[2]}.1.$cislo.$riadok
        posledny=$(snmpget -v1 -c ${config[1]} ${config[0]}
$tabla)
        hodnota=$(echo $posledny | cut -d ' ' -f4)

        case "$hodnota" in
            00) vypis="lowPaper" ;;
            01) vypis="noPaper" ;;
            02) vypis="lowToner" ;;
            03) vypis="noToner" ;;
            04) vypis="doorOpen" ;;
            05) vypis="jammed" ;;
            06) vypis="offline" ;;
            07) vypis="serviceRequested" ;;
            08) vypis="inputTrayMissing" ;;
            09) vypis="outputTrayMissing" ;;
```

```
    10) vypis="markerSupplyMissing" ;;
    11) vypis="outputNearFull" ;;
    12) vypis="outputFull" ;;
    13) vypis="inputTrayEmpty" ;;
    14) vypis="overduePreventMaint" ;;
    *) vypis="Neznama hodnota" ;;
esac

    echo -e -n "$vypis\t\t\n"
fi
else
    echo ""
    break 2
fi
done

((riadok++))
done
```

Po spustení programu sa vypíše daná tabuľka.