



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content
pentru învățământul superior tehnic

Utilizarea Sistemelor de Operare

32. Shell scripting

sed

- **sed -n `/abc/p`**
 - echivalent grep
- **sed -n `ş=`**
 - echivalent wc -l
- **sed `10q`**
 - echivalent head
- **Substitute**
 - **sed `s/ana/bogdan`**
 - **sed `s/ana/bogdan/g`**

sed (2)

- Stream editor
- Citește linie cu linie și face prelucrări
- Sintaxă spartană
 - range + comenzi
 - **\$=** (la sfârșitul fișierului, afișează numărul liniei)
 - **10q** (la linia 10, quit)
 - **s/ana/bogdan/g** (peste tot, substituie ana cu bogdan)
 - **1,10s/ana/bogdan/g** (în primele 10 linii, substituie)
 - **/corina/s/ana/bogdan/g** (pe liniile ce conțin corina, substituie)
- Sintaxa de substituție este folosită și în vi

awk

```
ifconfig | awk -F '\t]+' '{print $5;}'
```

```
dpkg -l 'apache2*' | grep ^ii | awk -F '\t]+'  
'{print $2;}'
```

```
< /etc/passwd awk -F ':' '{ if ($3 >= 1000) print  
$1;}'
```

awk (2)

- cut++
 - permite specificarea unei expresii regulate ca separator
- Limbaj de programare – sintaxă similară C
- Referă câmpurile prin variabile interne
 - \$0 – întreaga linie
 - \$1, \$2, ...
 - NR – number of records (număr de linii)
 - NF – number of fields (număr de coloane)

Recomandări shell scripting

- Nu folosiți shell scripting pentru ceea ce se face mai bine/usor/eficient în Python, C, Java, Ruby, PHP etc.
- Modularitate (do one thing, do one thing well)
- Nu reinventați roata
- Cleanup actions
- În general, nu folosiți majuscule în numele variabilelor
- Nu parsați ieșirea comenzii ls (folosiți stat, for i in *, find sau arrays)
- Folosiți ghilimele când referiți variabile
- <http://mywiki.woledge.org/BashFAQ>

Upload MAC address

```
#!/bin/bash
```

```
declare IFACE=eth1  
declare URL="http://anaconda.cs.pub.ro/~razvan/school/uso/lab/macs/putmac.php"  
declare MY_HWADDR
```

```
function getmac()  
{  
    MY_HWADDR=$(/sbin/ifconfig "$IFACE" | grep HW | awk -F '[\t]+' '{print  
    $5}')  
}
```

```
function uploadmac()  
{  
    wget -o /dev/null "$URL?mac=$MY_HWADDR" &> /dev/null  
}
```

```
function main()  
{  
    getmac  
    uploadmac  
}
```

```
main
```

```
exit 0
```

Run torrent script

```
#!/bin/bash

BTDOWNLOADHEADLESS=/usr/bin/btdownloadheadless

TORRENT_FILE="$1"
DOWNLOAD_DIR="$2"

function check_alive()
{
    ps -ef | grep $BTDOWNLOADHEADLESS | grep $TORRENT_FILE &> /dev/null
}

check_alive
if test $? -ne 0; then
    pushd . &> /dev/null
    cd $DOWNLOAD_DIR && nohup $BTDOWNLOADHEADLESS $TORRENT_FILE &> /dev/null &
    popd &> /dev/null
fi

exit 0
```


Mass add users

```
#!/bin/bash

IFS=","
INPUT_FILE=lists/cdl_2010_list.txt

while read username email name surname; do
    ./ldap_add $username $email $name $surname
done < $INPUT_FILE

exit 0
```

Migrare utilizatori

```
#!/bin/bash

#passwd_file  fisier cu useri gen /etc/passwd

while read user skel_pass uid gid comm home shell; do
    save_dir="/home/$user/old_anaconda"

    echo "$user $comm $home $shell"

    useradd -m -d /home/$user -g users -s $shell -c "${comm}" $user

    mkdir ${save_dir}
    chown $user:users ${save_dir}
    bash ./rsync.sh $home ${save_dir}

done < passwd_file
```

Send event

```
#!/bin/bash

# get subject
if test $# -ge 1; then
    subject="$1"
else
    echo -en "Subject: "
    read subject
fi

# get file
if test $# -ge 2; then
    file="$2"
else
    echo -en "File: "
    read file
fi

# send mail to all LDAP accounts
for i in $(ldapsearch -x mail | grep mail: | cut -d " " -f 2); do
    mail -s "$subject" $i < $file
done

exit 0
```

Wanna see something real scary?

```
#!/bin/sed -f

{
s/\([ \t]*http[^\s]+\)/[[\1]]/g
s/<pre>/<code>/g
s/<\/pre>/<\/code>/g
s/<source[ \t]*lang="[a-z]*"/<code \1>/g
s/<\/source>/<\/code>/g
s/<tt>/{{{/g
s/<\/tt>/}}}/g
s/<b>/**/g
s/<\/b>/**/g
s/<ul>/g
s/<\/ul>/g
s/[ \t]*<li>(.*)<\/li>/* \1/g
s/''/"/g
s/''/\\/g
s/{{{\\[[/[[[/g
s/\\]\\\}}}]}/]]/g
s/([ \t]*http:[^\s]+\)/\1|/g
}
```