

Sisteme de Operare

Laborator 2

Diana - Florina HALIȚĂ

Facultatea de Matematică și Informatică

Universitatea Babeș-Bolyai

<http://www.cs.ubbcluj.ro/~diana.halita>

S2

FILTRE UNIX

Filtre UNIX

Orice comanda care citește un fișier de la intrarea standard, îl transformă și îl afișează la ieșirea standard;

GREP

GREP

```
grep [ -chilnsvw ] [ [-e] expresie_regulară | -f nume_scenariu ] [ lista_fisiere ]
```

GREP - OPTIUNI

- **-c** - afiseaza numai numarul de linii care se potrivesc cu expresia regulara;
- **-h** - nu afiseaza numele fisierului;
- **-i** - ignore case
- **-l** - afiseaza numai numele de fisiere care contin sirul cautat, nu si liniile din aceste fisiere
- **-n** - afiseaza liniile care se potrivesc expresiei regulate precedate de numarul de ordine relativ la inceputul fiecarui fisier
- **-v** - afiseaza liniile care nu contin sirul dat
- **-w** - afiseaza liniile in care sirul dat este cuvant intreg
- **-e** - se foloseste daca vrem ca expresia regulara sa inceapa cu "-"

GREP - EXAMPLE

- `grep "ceva" fis`
- `grep -c "ceva" fis`
- `grep "ceva" fis f2` \Rightarrow la fiecare inceput de linie se specifica numele fisierului din care linia face parte
- `grep -h "ceva" fis f2` \Rightarrow nu se mai scrie numele fisierului
- `grep -l "ceva" fis f2 f3`
- `grep ^ if fis1 fis2` \Rightarrow afișează liniile din fis1 și fis2 care încep cu if
- `grep if fis1 fis2` \Rightarrow afișează toate liniile din fis1 și fis2 care conțin secvența "if"
- `grep fi$ fis1 fis2` \Rightarrow afișează liniile din fis1 și fis2 care se termină cu fi
- `grep '^\ (.* \) \ (.* \) \ 1 $' fis1` \Rightarrow afișează toate liniile care încep și se termină cu același cuvânt separator fiind în spațiu și conțin mai mult de două cuvinte

SED

SED

```
sed [-n] [-e scenariu ] [ lista_fisiere ] [-f fisier_scenariu ]
```

Scenariu: conditie instructiune

SED - CONDIȚII

- **condiție vidă** - adevărată pentru toate liniile din fișier;
- **n** - adevărată pentru linia cu numărul de ordine egal cu n (liniile se număratează cumulativ în lista de fișiere);
- **\$** - condiție adevărată pentru ultima linie din fișier;
- **/expresie regulată/** - condiție adevărată pentru liniile care conțin cel puțin un subsir care se potrivește cu expresia regulată;
- **expr1,expr2** - adevărată pentru liniile aflate între linia care se potrivește cu expr1 și linia care se potrivește cu expr2

SED - INSTRUȚIUNI

- **p** - afișează bufferul temporar la ieșirea standard;
- **d** - șterge zona temporară;
- **i \ < ENTER >** - are ca parametru un text (dat pe liniile următoare în fișierul scenariu) pe care îl afișează la ieșirea standard;
- **a \ < ENTER >** - analog cu **\ i** dar afișează după prelucrarea fiecărie linii
- **y/sir1/sir2/** - realizează o traducere înlocuind caracterele din fișierele de intrare care se găsesc în **sir1** cu caracterele corespunzătoare din **sir2** (unde **sir1** și **sir2** au lungimi egale);
- **s/expresie_regulara/sir/[flaguri]** - înlocuiește prima apariție a unui șir care se potrivește cu expresia regulată cu șirul "sir"

SED - INSTRUȚIUNI

Flaguri

- **nimic** - inlocuieste numai prima apariție;
- **număr n între 1 si 512** - inlocuieste exact a n-a apariție;
- **g** - inlocuieste toate aparițiile din linii;
- **p** - afișează buferul tampon la ieșire dacă s-a produs vreo modificare în linia respectivă

SED - EXAMPLE

- `echo Sunday | sed s/day/night/` ⇒ `Sunnight`
- `sed s/day/night old > new`
- `echo abcd123 | sed 's/ \ ([a-z]* \).*/ \ 1/'`
- `echo abcd123ef | sed 's/ \ ([a-z]* \).*/ \ 1/'` ⇒ afiseaza numai primul grup de litere
- `echo abcd efg | sed 's/ \ ([a-z]* \) \ ([a-z]* \)/ \ 2 \ 1/'` ⇒ interschimba primele 2 cuvinte
- `echo abcd efg | sed 's/ \ ([a-z]* \) \ ([a-z]* \)/ \ 1 \ 2 /'` ⇒ elimina spatiul dintre cuvinte
- `sed /DA/s/bada/banu/g old` ⇒ face inlocuirea numai in liniile care contin DA
- `echo "123def" | sed 'y/123456/ABCDEF/'`

AWK

AWK

```
awk [ -f fis_scenariu ] [ -Fc ] [ scenariu ] [ -v variabila=valoare... ]  
[fișiere...]
```

Scenariu: *conditie* {instructiuni}

- prelucreaza fișiere text selectand acele linii din text care satisfac anumite conditii;
- trateaza pe rand cate o linie din fișierele de intrare si pentru fiecare executa *instructiuni* atunci cand *conditie* ia valoarea *true*
- daca *conditie* lipsește atunci se executa instructiuni pentru toate liniile din *fișiere*

AWK

- conditii predefinite: BEGIN, END
- variabile predefinite: NF, NR, FNR, FS, FILENAME, OFS, ORS, ARGV, ARGC
- functii predefinite: length(s), substr(s,p,n), index(s1,s2), sprintf(format, arg1,...), split(s,a,c)
- Accesarea campurilor se face cu \$1, \$2 ...\$i, \$(i+1), \$NF, iar intreaga linie se referă cu \$0

AWK - EXAMPLE

- `awk '{print $1}' fis` \Rightarrow afiseaza dintr-un fisier primul cuvant din fiecare linie
- `awk -v v=anca '{print "Salut ", v}' fis` \Rightarrow afiseaza Salut anca de numarul de linii al fis
- `awk -F: '$2=="'" { print $5}' /etc/passwd` \Rightarrow afiseaza toti utilizatorii din sistem care nu au parola
- `{F[FILENAME]++}`
`END {for (f in F) print f, ":", F[f]}` \Rightarrow afiseaza numarul de linii pentru fiecare fisier prelucrat

AWK - EXAMPLE

- `{F[FILENAME]+=length($0)}`
`END {for (f in F) print f, ":", F[f]}` \Rightarrow afiseaza numarul de caractere pentru fiecare fisier prelucrat
- `{for(i=1; i<=NF; i++) X[$i]++} END {for (c in X) print "cuvantul", c, "apare de ", X[c], "ori!"}` \Rightarrow afiseaza toate cuvintele dintr-un fisier si numarul lor de aparitii