

Filters & Text Manipulation Tools

I. Filters

→ 1. Printing: `pr-sh`

```
% ls /home/cs476 | pr -h "--cs 476-" -l 20 -2 -n -d | more -20
```

-h header, **-l** page length, **-2** columns, **-n** add line numbers, **-d**

double space

→ 2. File Comparison: `compare-sh`

```
% paste f1 f2
```

```
a a
b b
c d
e
```

```
% cmp f1 f2
```

```
f1 f2 differ: char 5, line 3
```

```
% comm f1 f2
```

```
a
b
c
d
e
```

```
% diff f1 f2
```

```
3c3,4
< c
---
> d
> e
```

→ **3. Head & Tail:** **headtail-sh**

```
% cat f1
```

```
a  
b  
c
```

```
% head -n 1 f1
```

```
a
```

```
% tail -n 1 -f f1
```

```
c
```

/usr/bin/tail options: **-n** no of lines, **-f** force wait **-r** reverse lines

Exercise: getting the middle **middle-sh**

```
% head -n 1 f1 > h  
% tail -n 1 f1 > t  
% comm -23 f1 h > f1_h  
% comm -23 f1_h t > m
```

→ **4. Cut & Paste:** **cutpaste-sh**

```
% cat fields
```

```
Wahab, Hussein  
Maly, Kurt
```

```
% cut -d, -f2 fields > F
```

```
Hussein  
Kurt
```

-d delimiter, **-f** field number

```
% cut -d, -f1 fields > L
```

```
% cat L
```

```
Wahab
Maly
```

```
% paste -d" " FL > names
```

```
% cat names
```

```
Hussein Wahab
Kurt Maly
```

```
% cat names emails
```

```
Hussein Wahab
Kurt Maly
wahab@cs.odu.edu
maly@cs.odu.edu
```

```
% paste names emails
```

```
Hussein Wahab wahab@cs.odu.edu
Kurt Maly maly@cs.odu.edu
```

```
% cat info
```

```
Hussein Wahab
wahab@cs.odu.edu
x4512
Kurt Maly
maly@cs.odu.edu
x3915
```

```
% paste -s -d" ;\n" info > info2
```

-s concatenate separate lines, -d lines delimiters

```
% cat info2
```

```
Hussein Wahab;wahab@cs.odu.edu;x4512
Kurt Maly;maly@cs.odu.edu;x3915
```



5. Sort: sort-sh

```
% cat sortdata
```

```
Wahab,Hussein
Maly,Kurt
Wahab,Hussein
Maly,Kurt
```

```
% sort sortdata
```

```
Maly,Kurt
Maly,Kurt
Wahab,Hussein
Wahab,Hussein
```

```
% sort -u sortdata
```

```
Maly,Kurt
Wahab,Hussein
```

```
% sort -t, -k 2 sortdata
```

```
Wahab,Hussein
Wahab,Hussein
Maly,Kurt
Maly,Kurt
```

```
% paste f1 f2
```

```
a a
b b
c d
e
```

```
% sort -u -m f1 f2
```

```
a
b
c
d
e
```

-u unique, **-t** field separator, **-k** field number **-m** merge



6. Uniq: `uniq-sh`

% `cat grades`

```
1: A
2: B
3: C
4: D
5: F
6: A
7: A
8: B
9: F
```

% `cut -d: -f2 grades | sort | uniq -c`

```
3 A
2 B
1 C
1 D
2 F
```

-c count the occurrence of each value



7. Translate: `translate-sh`

% `cat fields | tee /dev/tty | tr 'a-z' 'A-Z' | tee /dev/tty | tr 'A-Z' 'a-z'`

```
Wahab, Hussein
Maly, Kurt
WAHAB, HUSSEIN
MALY, KURT
wahab, hussein
maly, kurt
```



8. Egrep: `egrep-sh`

```
% ls -l /home/cs476 | egrep "^d.*[M|m]ail.*"
```

```
drwx----- 2 cs476 cs476 512 Oct
22 1999 Mail
drwx----- 2 cs476 cs476 512 Dec
9 2003 mail
drwx----- 2 cs476 cs476 512 Nov
3 1998 nsmail
```

```
% vpcat passwd | egrep "^cs[5-9][0-9]+" | cut -d: -f1 | sort -u | tee /dev/tty | wc
-l
```

```
cs554
cs555
cs558
cs588
cs656
cs695
cs745
cs772
cs775
cs778
cs779
cs845
```

12



9. sed: sed.sh

```
% cat grades
```

```
1: A
2: B
3: C
4: D
5: F
6: A
7: A
8: B
9: F
```

```
% sed "s/^/000/; s/A/Excellent/; s/B/Very Good/; s/C/Good/; s/D/Pass/;
s/F/Fail/" grades
```

```
0001: Excellent
0002: Very Good
0003: Good
0004: Pass
0005: Fail
0006: Excellent
0007: Excellent
0008: Very Good
0009: Fail
```

% sed "1d; \$d" grades

```
2: B
3: C
4: D
5: F
6: A
7: A
8: B
```

Exercise: getting the middle **midsed-sh**



10. Join: **join-sh**

% more fac

```
Hussein Wahab:1:Networks
Kurt Maly:1:Digital Lib
Steve Olariu:1:Theory
Mike Oversteet:2:Software Engineering
Jessica Crouch:3:Graphics
Chris Wild:4:Artificial Intellegence
Ravi Mukkamala:1:Distributed Systems
Mohammed Zubair:1:Digital Lib
Irwin Levenstien:2:Databases
Stewart Shen:1:Web Technolgy
```

% more rank

```
3:Assistant Professor:starting rank
4:Emiratus Professor: Retired
1:Professor top rank
2:Assocaite Professor: middle rank
```

% more join_result

```
1:Hussein Wahab:Networks:Professor: top rank
1:Kurt Maly:Digital Lib:Professor: top rank
1:Mohammed Zubair:Digital Lib:Professor: top rank
1:Ravi Mukkamala:Distributed Systems:Professor: top rank
```

```

1:Steve Olariu:Theory :Professor: top rank
1:Stewart Shen:Web Technolgy:Professor: top rank
2:Irwin Levenstien:Databases:Assocaite Professor:
middle rank
2:Mike Oversteet:Software Engineering:Assocaite
Professor: middle rank
3:Jessica Crouch:Graphics:Assistant Professor:starting
rank
4:Chris Wild:Artificial Intellegence:Emiratus
Professor: Retired

```

% more join-sh

```

sort -t: -n -k2 fac > facS
sort -t: -n -k1 rank > ranks
join -t: -1 2 -2 1 facS ranks > join_result.all
join -o 1.1 2.2 1.3 -t: -1 2 -2 1 facS ranks > join_result.some

```

-n numeric

-1 2 -2 1 join fields of first and second files

-o fields of file1 or file2

II. AWK

→ **1. Selecting and printing lines:** `ex1-sh`

CS courses accounts: print login and name:

```
% ypcat passwd | awk -F: ' /^cs[5-9][0-9]+/ { print $1, $5 } '
```

```

cs656 cs656 Class Account
cs554 ajay's grad Networking class
cs775 CS775 Grader Account

```

-F Field separator

Faculty accounts: print login and name:

```
% ypcat passwd | awk -F: ' $4 == 13 { print $1 "-->" $5 } '
```

```

Shen --> Stewart Shen
Wahab --> Dr. wahab

```

Faculty accounts: print all fields except the password field:

```
% ypcat passwd | awk -F: ' $4 == 13 { $2= "" ; {print} } '
```

```
shen 55 13 Stewart Shen /home/shen
/usr/local/bin/tcsh
wahab 51 13 Dr. wahab /home/wahab
/usr/local/bin/tcsh
```

Faculty accounts: print line number, login and name:

```
% ypcat passwd | awk -F: ' $4 == 13 { print ++count ": " $1 "->" $5 } '
```

```
5: shen --> Stewart Shen
11: Wahab --> Dr. Wahab
```

Exercise: getting file name from a path **pathToFile-sh**

```
echo $1 | awk -F/ '{print $NF}'
```

→ **2. awk file:** **ex2-sh** & **ex2-awk**

```
ex2-sh: ypcat passwd | awk -F: -f ex2-awk
```

```
ex2-awk: $4 == 13 { print ++count, $1 "-->" $5 }
```

-f file containing instructions

→ **3. Begin and END:** **ex3-sh** & **ex3-awk**

```
ex3-sh: ypcat passwd | awk -F: -f ex3-awk
```

```
ex3-awk:  
BEGIN |
```

```
system ("date");
printf " HOME = %s \n", ENVIRON["HOME"]
```

```
$4 == 13 { print ++count, $1 "--> " $5 }
```

```
END |
```

```
printf "Total number is %d\n", count
system ("who");
printf "PATH = %s \n", ENVIRON["PATH"]
```

III. PERL

→ 1. Selecting and printing lines: `ex1pl-sh` & `ex1-pl`

```
ex1pl-sh: ypcat passwd | ex1-pl
```

```
ex1-pl:
```

```
#!/usr/bin/perl
system ("date");
print ("BEGIN - finding cs accounts \n");
```

```
while (<>) {
    if (/^cs[4-9][0-9]+/) {
        $count++;
        print ("$. --> $_ ");
    }
}
```

```
print ("END - total: $count \n");
```

\$. Current line number, \$_ Content of current line.

→ **2. Selecting fields:** `ex2pl-sh` & `ex2-pl`

```
ex2pl-sh: ypcat passwd | ex2-pl
```

```
ex2-pl:
```

```
#!/usr/bin/perl
```

```
print ("ENV-HOME: $ENV{'HOME'} \n");  
print("BEGIN - finding cs accounts \n");
```

```
$lineno = 0 ;  
while (<>) {  
    if (/^cs[4-9][0-9]+/) {  
        split (/:/);  
        $count++;  
        print ("$. --> $_[0] $_[4] \n " );  
    }  
}
```

```
print ("END - total: $count \n");
```

`$_[i]` the ith field

Exercise: getting file name from a path

```
path2file-sh:
```

```
echo $1 | path2file-pl
```

```
path2file-pl:
```

```
#!/usr/bin/perl
```

```

while (<>){
    $nf = @fields = split (/\/);
    print (@fields[$nf-1]);
}

```

→ **3. Group count: ex3-pl**

ex3-pl:

```
#!/usr/bin/perl
```

```

open(INFILE, "grades");
print("BEGIN - counting\n");

```

```

while (<INFILE> ) {
    chomp(); //remove end of line
    split (/:/);
    $grade = $_[1] ;
    $gradelist{$grade}++ ;
}

```

```

foreach $grade (sort ( keys %gradelist ) ) {
    print (" $grade ---> $gradelist{$grade} \n") ;
}

```

```
print ("END - grade list: \n");
```

→ **4. Translate & substitute: tr-pl & substitute-pl**

tr-pl:

```
#!/usr/bin/perl
```

```
open(INFILE, "fields");  
print("BEGIN - translate \n");
```

```
while (<INFILE>) {  
    tr /a-z/A-Z/;  
    print;  
}
```

```
print ("END - list: \n");
```

substitute-pl:

```
#!/usr/bin/perl  
open(INFILE, "grades");  
print("BEGIN - substitute \n");
```

```
while (<INFILE>) {  
    s/^/000/; s/F/Fail/; s/A/Excellet/ ; s/B/Very Good/ ; s/C/Good/ ;  
    s/D/Pass/ ;  
    print;  
}
```

```
print ("END - list: \n");
```