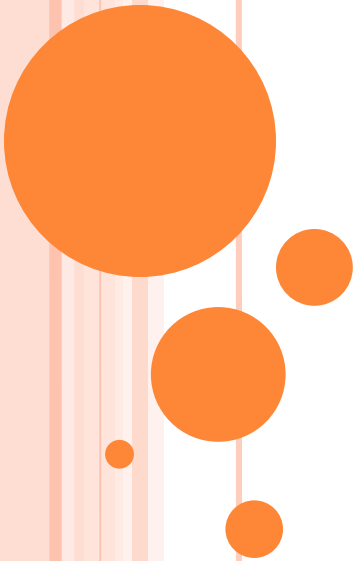
CS5354
UNIX TOOL PROGRAMMING



Filters

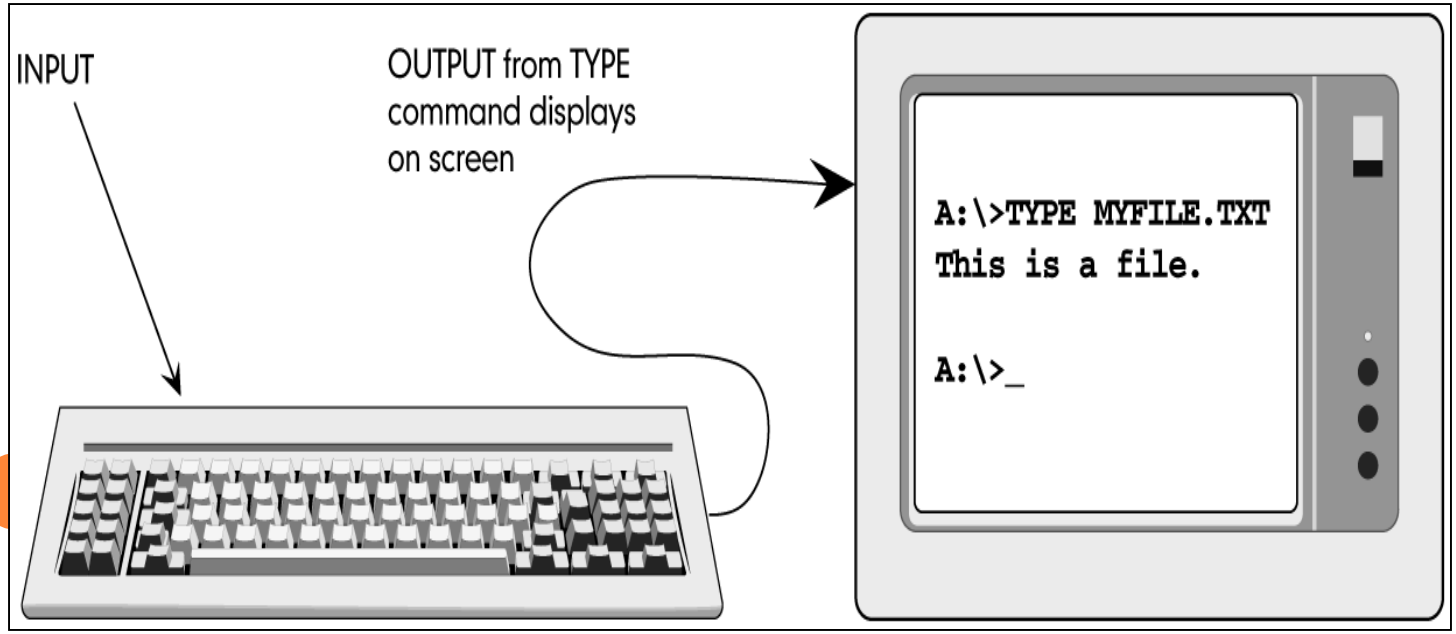
- A. Sawarkar

Redirection of Standard I/O (Input/Output)

Operating system gets information from or sends information to:

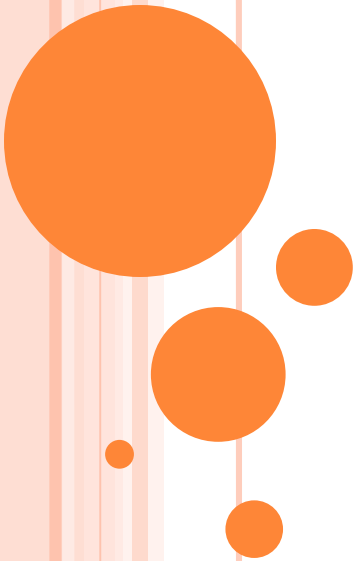
- ➔ Standard input
- ➔ Standard output
- ➔ Standard error

Redirection of Standard I/O (Input/Output)

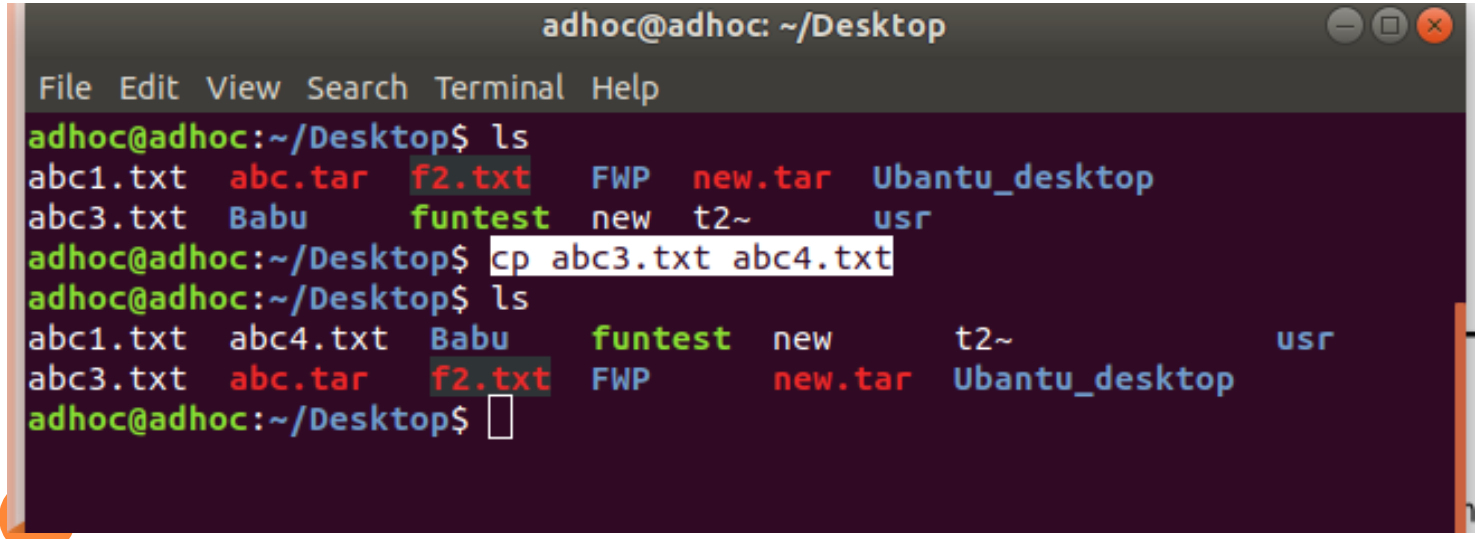


Redirection of Standard I/O (Input/Output)

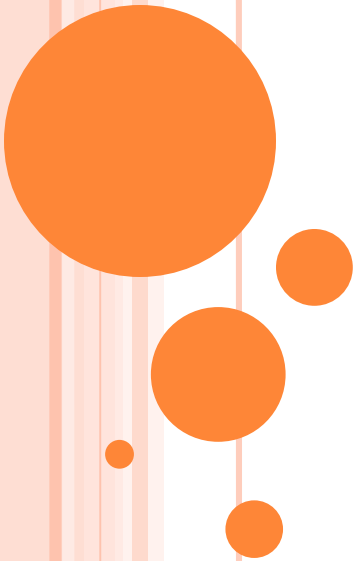
Not all commands deal with standard input/output



Redirection of Standard I/O (Input/Output)



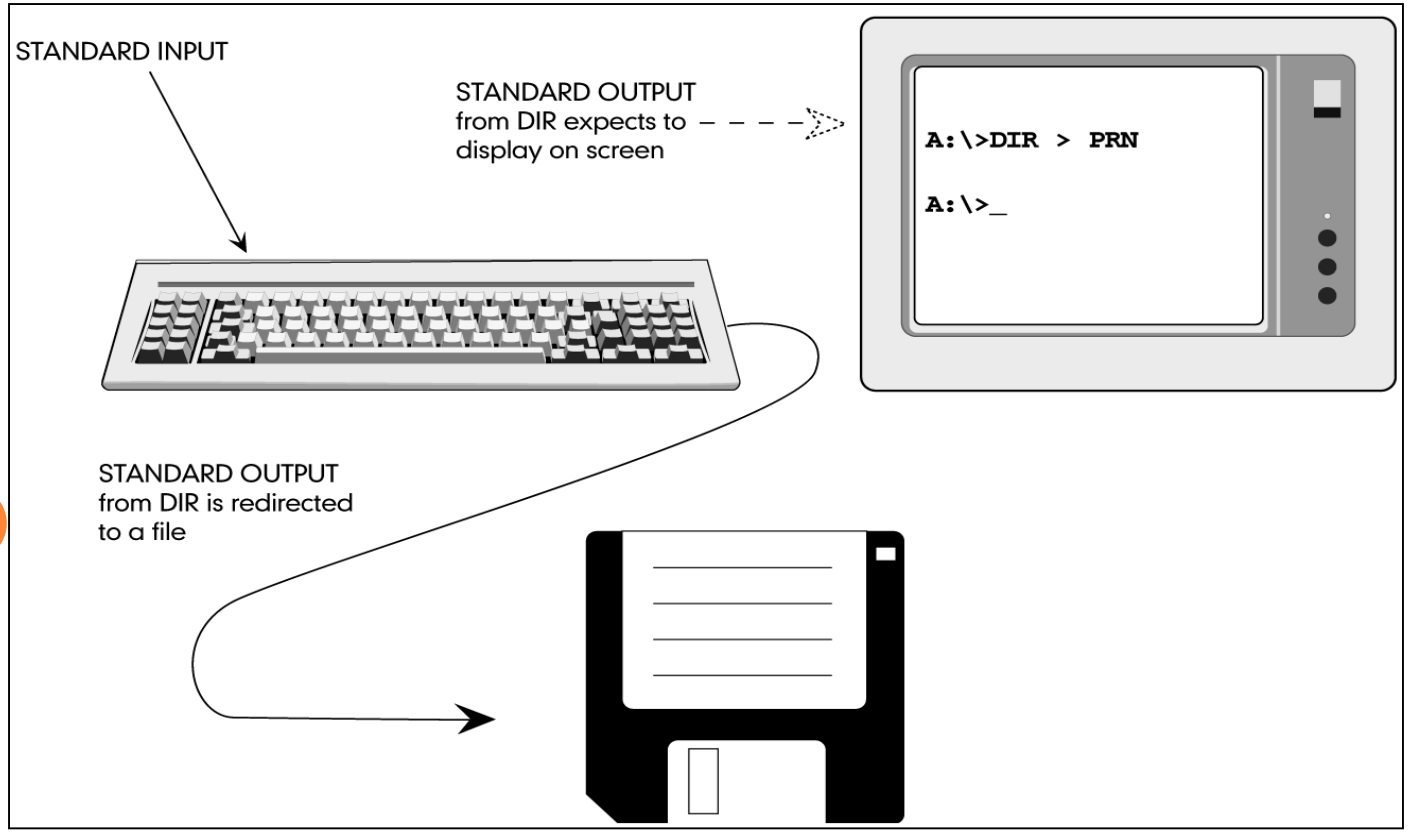
Redirection of Standard I/O (Input/Output)



I/O redirection tells OS ,you want information read from or written to a device other than the

standard ones.

Redirection of Standard I/O (Input/Output)



Redirection of Standard I/O (Input/Output)

Symbols used for redirection:

> (greater than: Output)

< (less than: Input)

>> (double greater than:Append)

```
$ bc < calc.txt  > result.tx
```

Input

Output

Unix commands :4 categories

1. **Directory oriented**

   --- mkdir, rmdir, cd, cp, mv, rm

2. **Output**

   – Commands don't read std input but they write to **std output**

   – ls, pwd, who, etc

3. **Input**

   – Commands that read **std input** but don't write to std output

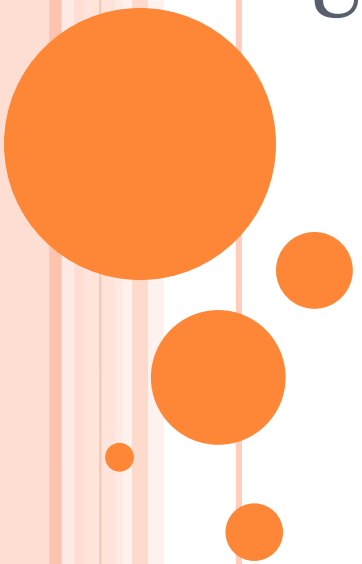
   – lp

4. **Input + Output**

   – Commands both **std input and std output**

   – cat, wc,cm,gzip

The 4th category are called Filters in UNIX



Filter History

- Originally conceived by M. D. McIlroy in the early 1970's
- The UNIX notation for pipeline, the vertical bar, was introduced by K. L. Thompson.
  - "The Unix time-sharing system," *Comm. ACM,* July 1974.
  - "The Unix programming environment," *Software Practice and Experience,* January 1979.

CONTENTS

Filters – definition
To format text – **pr**
Pick lines from the beginning – **head**
Pick lines from the end – **tail**
Extract characters – **cut**
Join two lines / files – **paste**
Sort, merge and remove – **sort**
Find unique and no unique lines – **uniq**
Change, delete or squeeze characters - **tr**

Filters

- Definition: A *filter* is a program that takes a text file as an input and produces a text file as an output.
- UNIX context
  - Write filters to use *stdin* as the input file and *stdout* as the output file.
  - Use *pipe* to connect filters. Notation is the vertical bar |

Filter Examples

Calc.txt

```
2^32
25*50
30*25+15^2
```

```
$ bc < calc.txt  > result.txt
$ cat result
  4294967296
    1250
     975
```

Output

Input

Note: Here sequence can be anything as

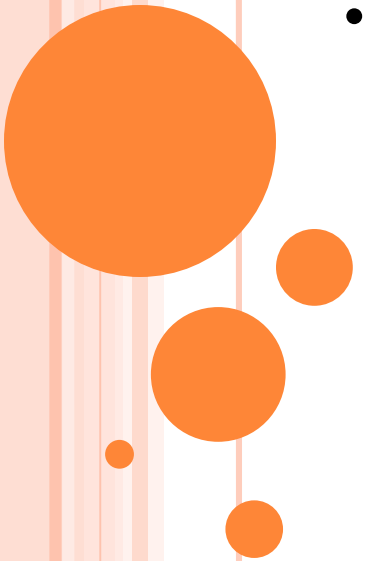
1. bc > result.txt <calc.txt
2. > result <calc.txt bc

SIMPLE FILTERS

- Commands which accept data from **standard input**, manipulate it and write the results to **standard output**
- Each filter performs a simple function
- Some commands use delimiter, pipe (|) or colon (:)
- Many filters work well with delimited fields, and some simply won't work without them

- The piping mechanism allows the standard output of **one filter** serve as standard input of **another**

- The filters can read data from standard input when used without a filename as argument, and from the file otherwise



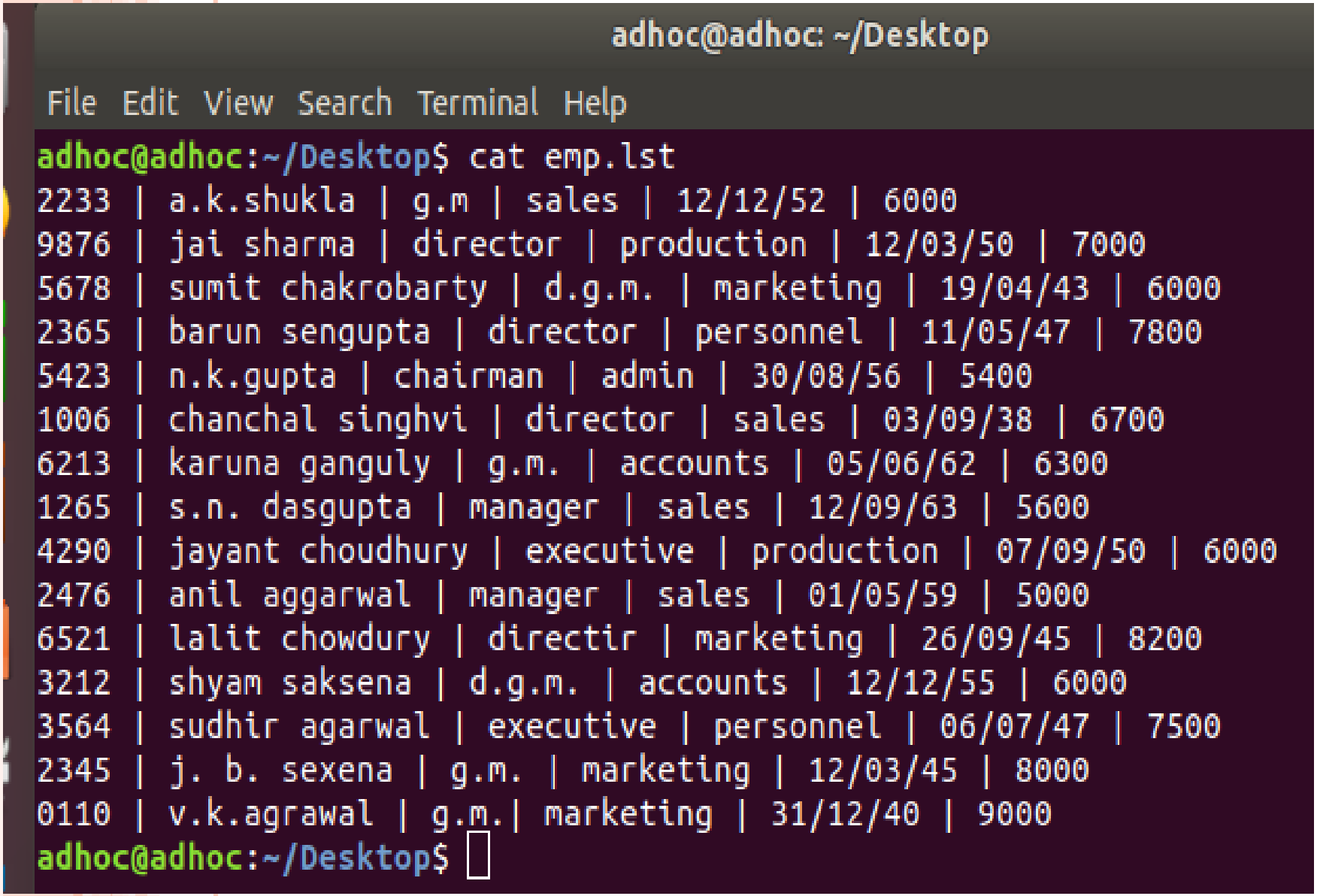
THE SIMPLE DATABASE

- $ head emp.lst

```
2233 | shukla  | g.m    | sales   | 12/12/52 | 20000
9876 | sharma |d.g.m  |product| 12/03 60 | 15000
7898 | akash   |dir.     |mark.  | 11/06/70 |9000
3456 | tiwary   |g.m     |product| 05/02/89 |23000
1234 | kumar  | mgr    |accnts   | 18/03/79 |15000
3456 | anil      |chman  |sales    | 30/02/69 |40000
6789 | lalith    |mrg     | mark.  | 17/01/80 |60000
5678 | a         | d       | m       | 12/12/80 |12000
```

This is the emp
database which stores

THE SIMPLE DATABASE



THE SIMPLE DATABASE

- Several UNIX commands are provided for text editing and shell programming (emp.lst)
- Each line of this file has six fields separated by five delimiters
- The details of an employee are stored in one single line

2233 | a.k.shukla | g.m | sales | 12/12/52 | 6000

Emp_id | Name | Designation | Dept | DOB | Salary

pr :PAGINATING FILES

Pr command prepare a file for printing by adding suitable header , footers and formatted text.

- **pr adds five lines of margin at the top and bottom**
- The header shows the date and time of last modification of the file along with the filename and page number

We know that,

$ pr dept.lst

May 06 10:38 2017 dept.lst page 1

01 | accounts | 6213
02 | admin | 5423
03 | marketing | 6521
04 | personnel | 2365
05 | production | 9876
06 | sales | 1006

pr options

-k prints k (integer) columns

-t to suppress the header and footer

-h to have a header of user's choice

-d double spaces input

-n will number each line and helps in debugging

-o *n* offsets the lines by n spaces and increases left margin of page

If a program outputs a series of 20 numbers , one in each line then this option looks good.
-t: suppress the header and footer

$ a.out | pr –t -5

0	4	8	12	16
1	5	9	13	17
2	6	10	14	18
3	7	11	15	19

pr +10 chap01

starts printing from page 10

pr -l 54 chap01

this option sets the page length to 54

Simple Filters: **head** and **tail**

- Use **`head`** to look at the first few lines of a file
  - Keeps the first *n* lines of input and discards the rest

    `head` [-*n*] [*file*]
- Use **`tail`** to look at the last few lines of a file
  - Keeps the last *n* lines of input and discards the rest

    `tail` [-*n*] [*file*]
  - Discards the first *n*-1 lines of input and keeps the rest

    `tail` [+*n*] [*file*]

head

- Displays the top of the file
- It displays the first 10 lines of the file, when used without an option
- head emp.lst
- -n to specify a line count
- head -n 3 emp.lst

tail

- Displays the end of the file
- It displays the last 10 lines of the file, when used without an option
- tail emp.lst
- -n to specify a line count
- tail -n 3 emp.lst
- Monitoring the file growth (-f)

[seen log file at time of installation e.g oracle]

- Extracting bytes rather than lines (-c)

cut

- It is used for splitting the file vertically

$ head -n 5 emp.lst | tee shortlist

tee facility that save the output in shortlist and also display it on the terminal

#will select the first five lines of emp.lst and saves it to *shortlist*

- We can cut by using -c option with a list of **column** numbers, delimited by a comma (cutting columns)

$cut -c 6-22,24-32 shortlist

$cut -c -3,6-22,28-34,55- shortlist

- Most files don’t contain fixed length lines, so we have to cut fields rather than columns (cutting fields)

    -d for the field delimiter

    -f for the field list

$cut -d \ | -f 2,3 shortlist | tee cutlist1

will display the second and third columns of *shortlist* and saves the output in *cutlist1*. here | is escaped to prevent it as pipeline character

Cut examples

Appar
Hemant
Lucky
pankaj

$ cut [option] file

Option: -b(byte) ,-d(delimiter), -c(column)

1.$cut –b 1,2,3 file1.txt

[only show the first 3 column]

2.$cut –b 1-3,5-7 file1.txt

[Range : show byte 1 to 3 and 5 to 7]

3. $cut –c 2,5,7 file1.txt

[only show column 2,5 and 7]

pr
en
uy
aa

Cut examples…..

appar
hemant
lucky
pankaj

$ cut [option] file

Option: -b(byte) ,-d(delimiter), -c(column)

4. $cut –d “ ” –f 1 file1.txt

[delete space only one or show without space]

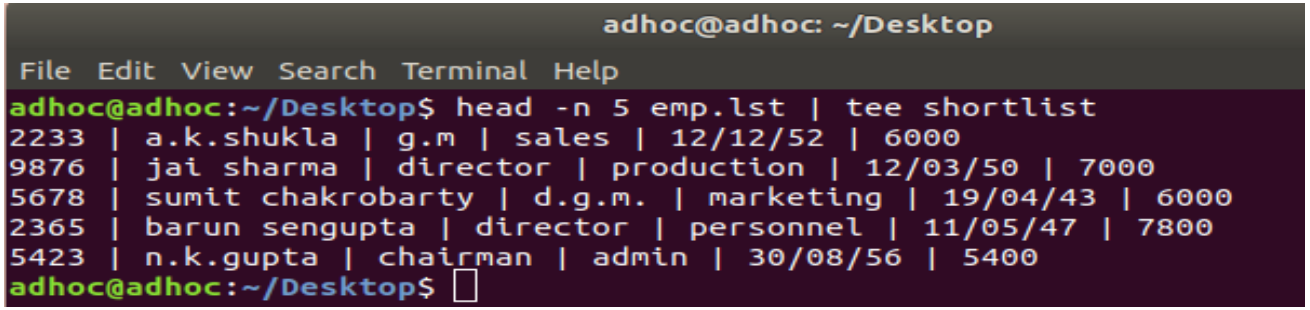
5. $cut –d “a” –f 1 file1.txt

[show details without “a” ..]

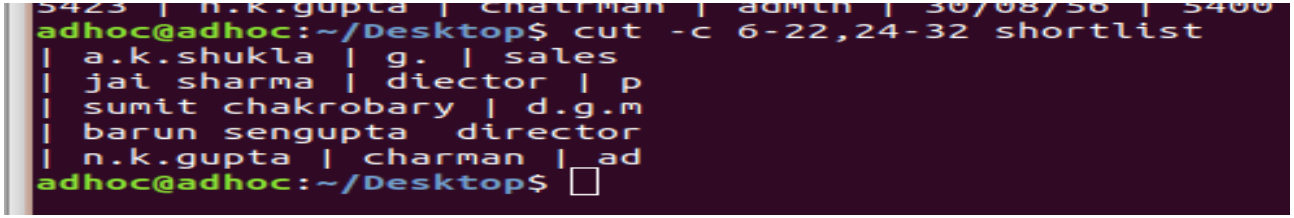
hem
Lucky
p

Cut examples…..

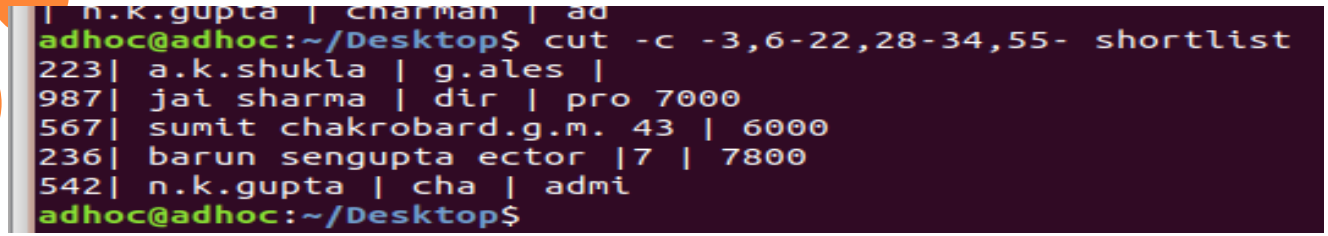
$head -n 5 emp.lst | tee shortlist



$cut -c 6-22,24-32 shortlist



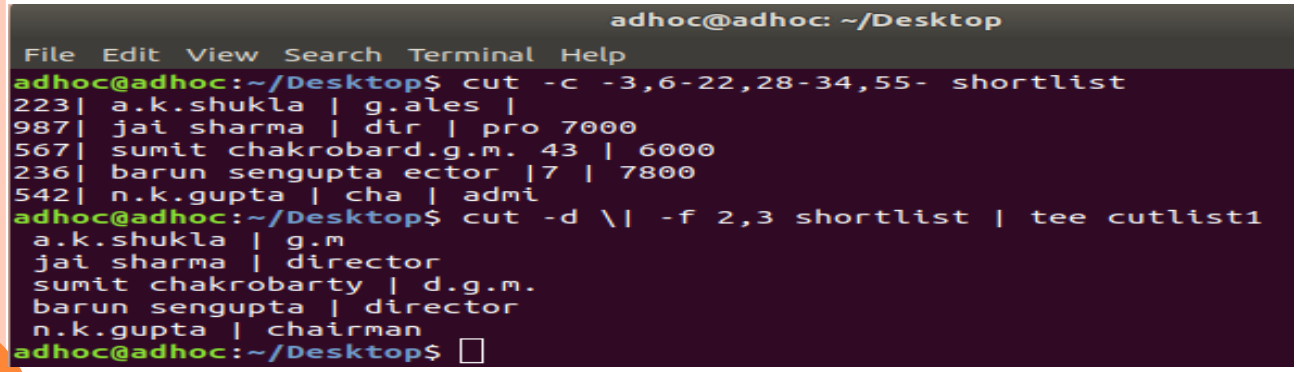
$cut -c -3,6-22,28-34,55- shortlist



The expression 55- indicates column number 55 to end of line. Similarly, -3 is the same as 1-3.

Cut examples…..

$cut -d \| -f 2,3 shortlist | tee cutlist1



display the second and third columns of *shortlist* and saves the output in *cutlist1*. here | is escaped to prevent it as pipeline character

To print the remaining fields, we have

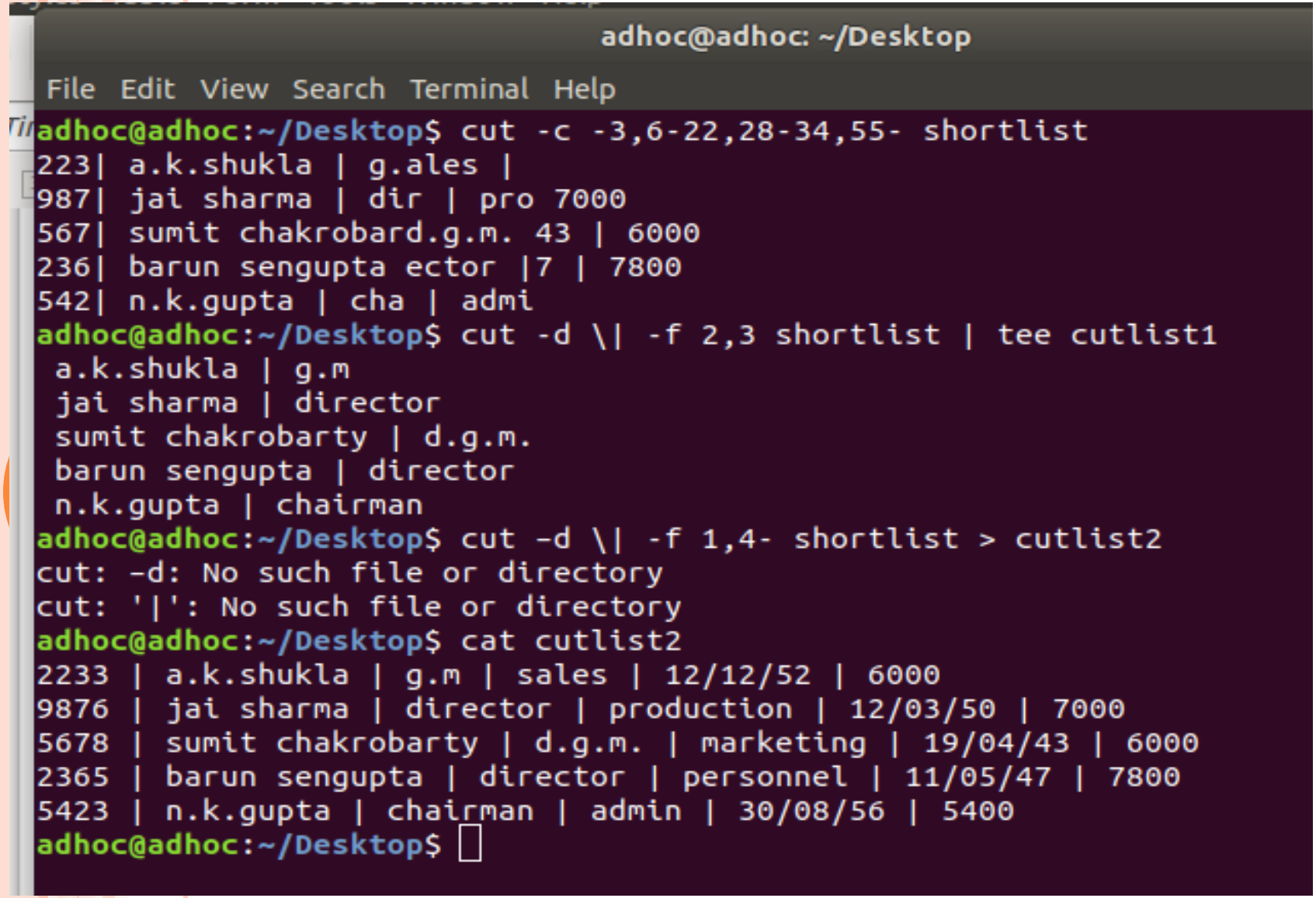
$cut –d \| -f 1,4- shortlist > cutlist2

```
adhoc@adhoc:~/Desktop$ cat cutlist2
2233 | a.k.shukla | g.m | sales | 12/12/52 | 6000
9876 | jai sharma | director | production | 12/03/50 | 7000
5678 | sumit chakrobarty | d.g.m. | marketing | 19/04/43 | 6000
2365 | barun sengupta | director | personnel | 11/05/47 | 7800
5423 | n.k.gupta | chairman | admin | 30/08/56 | 5400
adhoc@adhoc:~/Desktop$
```

Paste

When we cut with ***cut command,*** it can be pasted back with the ***paste*** command, *vertically rather than horizontally*

Input for Paste



Paste

When we cut with ***cut,*** it can be pasted back with the ***paste*** command, *vertically rather than horizontally*

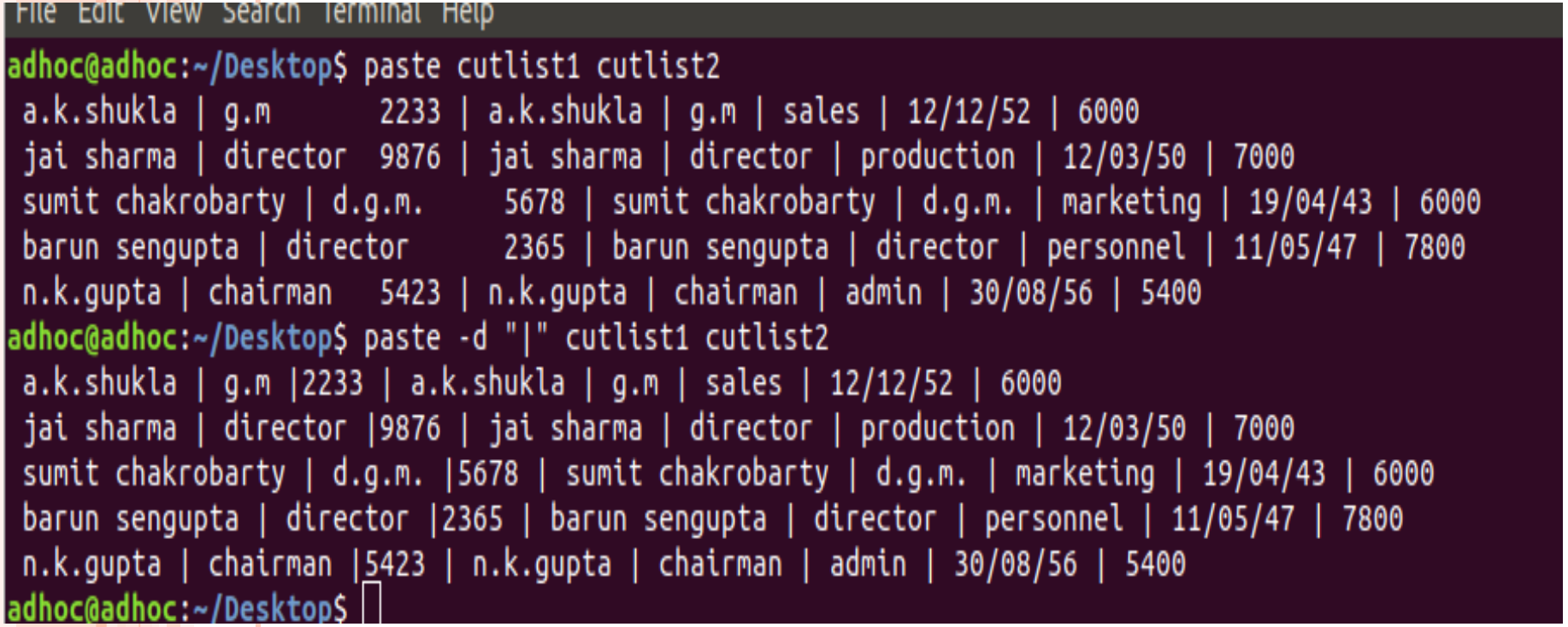
$ paste cutlist1 cutlist2

We can view two files side by side

```
adhoc@adhoc:~/Desktop$ cut -d \| -f 2,3 shortlist | tee cutlist1
 a.k.shukla | g.m
 jai sharma | director
 sumit chakrobarty | d.g.m.
 barun sengupta | director
 n.k.gupta | chairman
adhoc@adhoc:~/Desktop$ cut -d \| -f 1,4- shortlist > cutlist2
cut: -d: No such file or directory
cut: '|': No such file or directory
adhoc@adhoc:~/Desktop$ cat cutlist2
2233 | a.k.shukla | g.m | sales | 12/12/52 | 6000
9876 | jai sharma | director | production | 12/03/50 | 7000
5678 | sumit chakrobarty | d.g.m. | marketing | 19/04/43 | 6000
2365 | barun sengupta | director | personnel | 11/05/47 | 7800
5423 | n.k.gupta | chairman | admin | 30/08/56 | 5400
adhoc@adhoc:~/Desktop$ paste cutlist1 cutlist2
 a.k.shukla | g.m       2233 | a.k.shukla | g.m | sales | 12/12/52 | 6000
 jai sharma | director  9876 | jai sharma | director | production | 12/03/50 | 7000
 sumit chakrobarty | d.g.m.     5678 | sumit chakrobarty | d.g.m. | marketing | 19/04/43 | 6000
 barun sengupta | director      2365 | barun sengupta | director | personnel | 11/05/47 | 7800
 n.k.gupta | chairman   5423 | n.k.gupta | chairman | admin | 30/08/56 | 5400
```

We can specify one or more delimiters with -d

$paste -d “|” cutlist1 cutlist2



sort

Sorting is the ordering of data in ascending or descending sequence. The sort command orders a file and by default, the entire line is sorted

```
n.k.gupta | chairman |5423 | n.k.gupta | chairman | admin | 30/08
adhoc@adhoc:~/Desktop$ sort shortlist
2233 | a.k.shukla | g.m | sales | 12/12/52 | 6000
2365 | barun sengupta | director | personnel | 11/05/47 | 7800
5423 | n.k.gupta | chairman | admin | 30/08/56 | 5400
5678 | sumit chakrobarty | d.g.m. | marketing | 19/04/43 | 6000
9876 | jai sharma | director | production | 12/03/50 | 7000
adhoc@adhoc:~/Desktop$
```

This default sorting sequence can be altered by using certain options. We can also sort one or more keys (fileds) or use a different ordering rule.

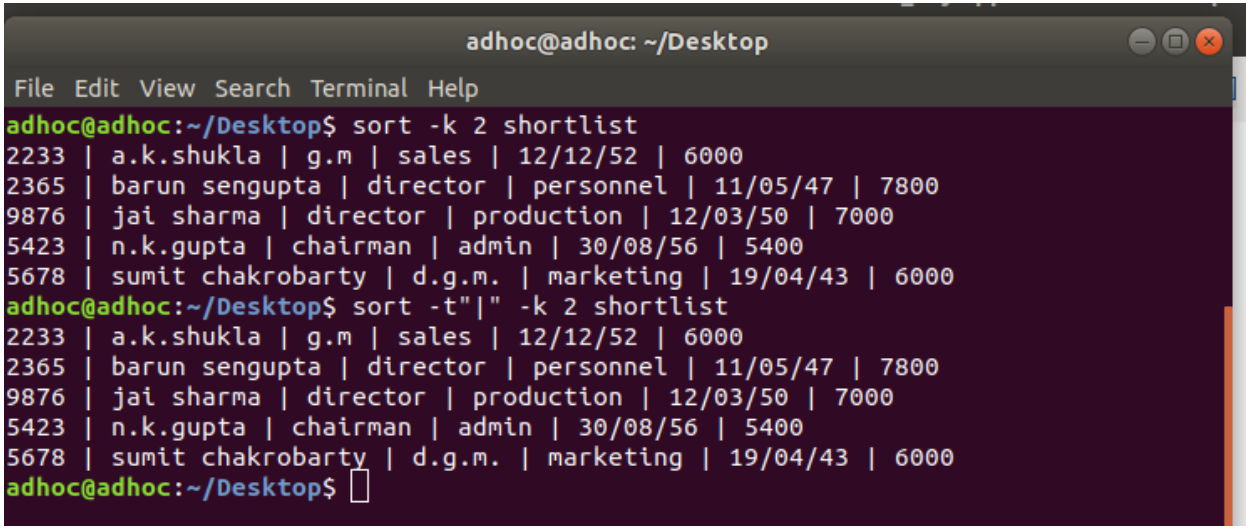
sort options

The important sort options are:

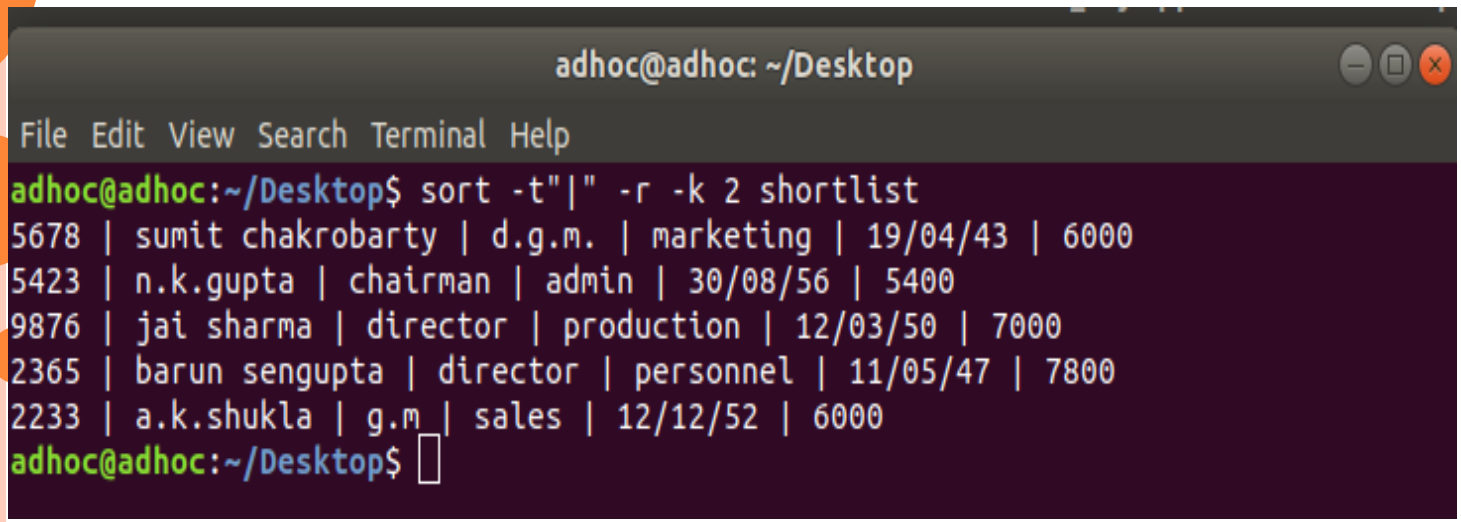
-t char	uses delimiter *char* to identify fields
-k n	sorts on nth field
-k m,n	starts sort on mth field and ends sort on nth field
-k m.n	starts sort on nth column of mth field
-u	removes repeated lines
-n	sorts numerically
-r	reverses sort order
-f	folds lowercase to equivalent uppercase
-m list	merges sorted files in list
-c	checks if file is sorted

Sort Examples

$sort –t“|” –k 2 shortlist



$sort -t “|” -r -k 2 shortlist



Sorting on secondary key

sort -t “|” -k 3,3 -k 2,2 shortlist

```
adhoc@adhoc:~/Desktop$ sort  shortlist
2233 | a.k.shukla | g.m | sales | 12/12/52 | 6000
2365 | barun sengupta | director | personnel | 11/05/47 | 7800
5423 | n.k.gupta | chairman | admin | 30/08/56 | 5400
5678 | sumit chakrobarty | d.g.m. | marketing | 19/04/43 | 6000
9876 | jai sharma | director | production | 12/03/50 | 7000
adhoc@adhoc:~/Desktop$ sort -t"|" -k 3,3 -k 2,2 shortlist
5423 | n.k.gupta | chairman | admin | 30/08/56 | 5400
5678 | sumit chakrobarty | d.g.m. | marketing | 19/04/43 | 6000
2365 | barun sengupta | director | personnel | 11/05/47 | 7800
9876 | jai sharma | director | production | 12/03/50 | 7000
2233 | a.k.shukla | g.m | sales | 12/12/52 | 6000
```

If 3rd field match used 2nd filed

Sorting on column

$sort -t"|" -k 5.7,5.8 shortlist

filed number 5 column 7 and 8 (DOB)

```
5678 | sumit chakrobarty | d.g.m. | marketing | 19/04/43 | 6000
2365 | barun sengupta | director | personnel | 11/05/47 | 7800
9876 | jai sharma | director | production | 12/03/50 | 7000
2233 | a.k.shukla | g.m | sales | 12/12/52 | 6000
5423 | n.k.gupta | chairman | admin | 30/08/56 | 5400
```

Sort on numeric

numfile

27
2
4
10

$sort numfile

10

2

27

4

$sort -n numfile

2

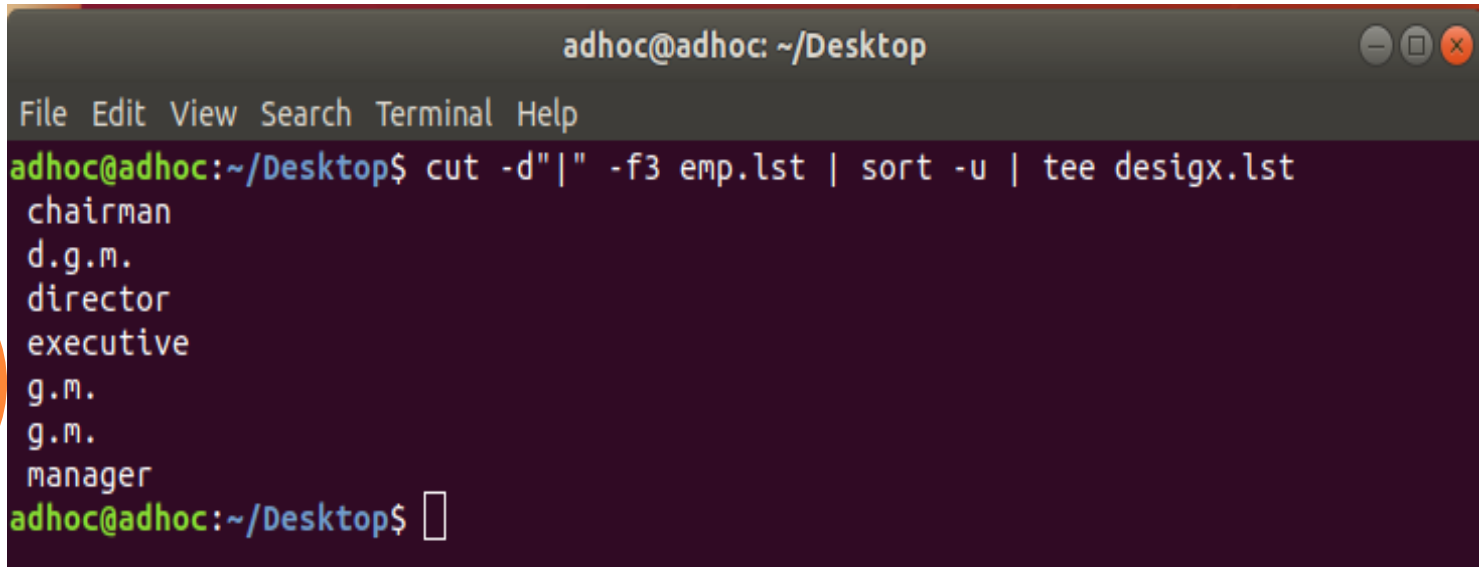
4

10

27

Sort and cut

$cut -d”|” -f3 emp.lst | sort -u |tee desigx.lst



Sort option

$sort -o sortted_list -k 3 shortlist

[output stored in sortted_list]

$Sort -o shortlist shortlist

[output stored in same file]

$sort -c shortlist

Uniq: Locate repeated and nonrepeated lines

$cat dept.lst

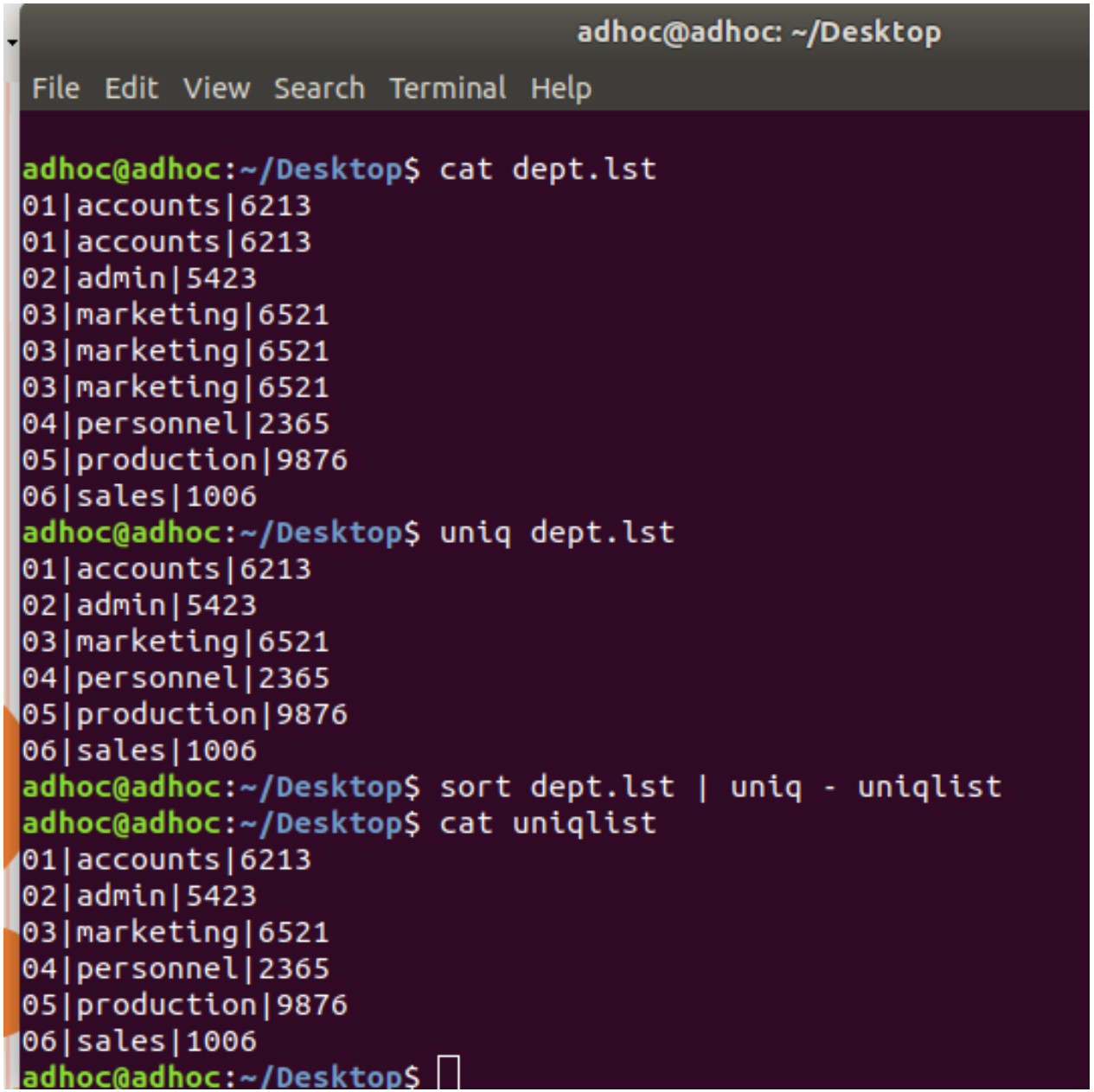
[show all duplicate entries of dept.lst]

$ uniq dept.lst

[Fetches one copy of each line and write it to std output]

$sort dest.lst |uniq -uniqlist

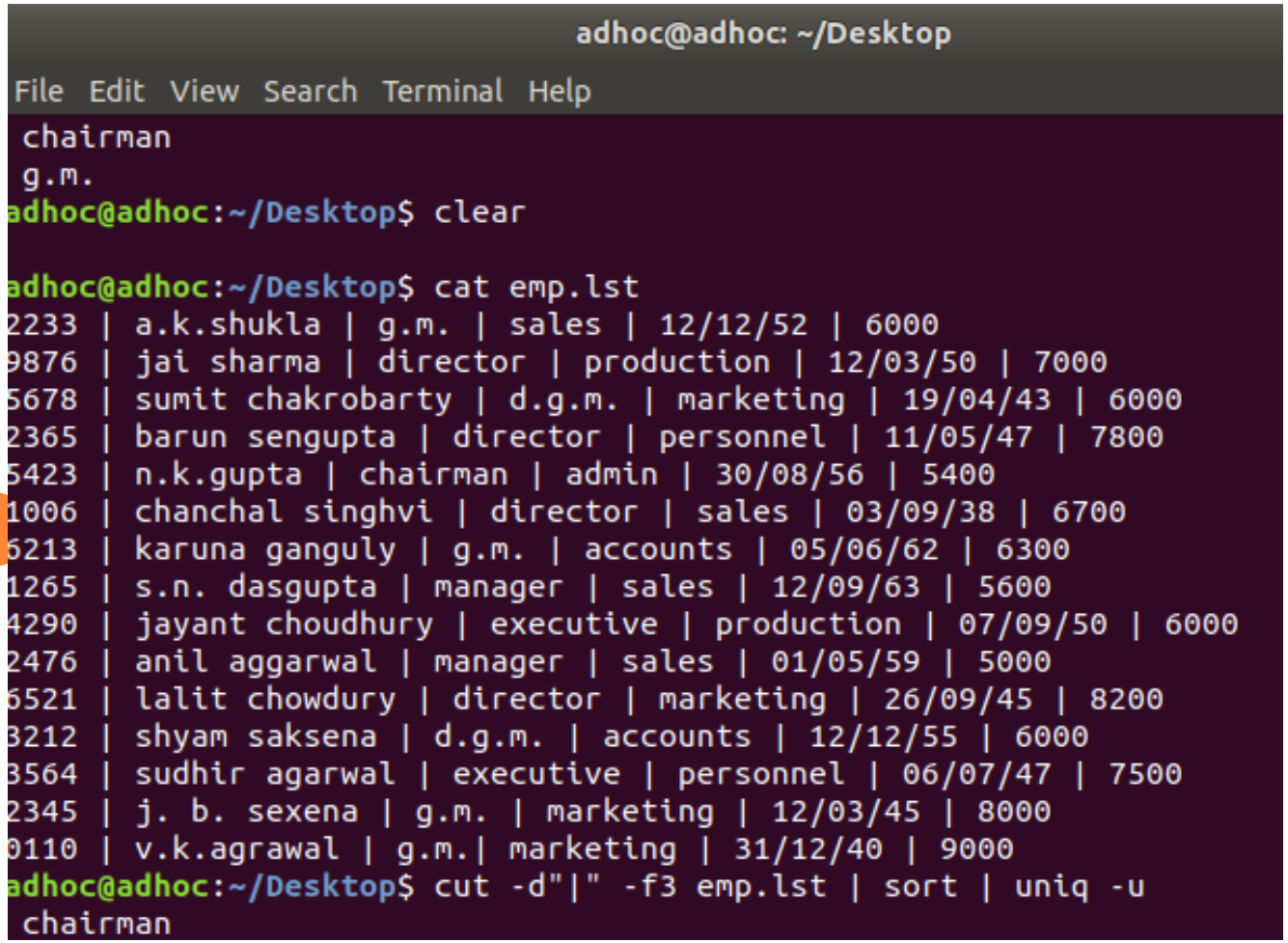
[using pipe its show same output]


adhoc@adhoc: ~/Desktop
File Edit View Search Terminal Help
adhoc@adhoc:~/Desktop$ cat dept.lst
01|accounts|6213
01|accounts|6213
02|admin|5423
03|marketing|6521
03|marketing|6521
03|marketing|6521
04|personnel|2365
05|production|9876
06|sales|1006
adhoc@adhoc:~/Desktop$ uniq dept.lst
01|accounts|6213
02|admin|5423
03|marketing|6521
04|personnel|2365
05|production|9876
06|sales|1006
adhoc@adhoc:~/Desktop$ sort dept.lst | uniq - uniqlist
adhoc@adhoc:~/Desktop$ cat uniqlist
01|accounts|6213
02|admin|5423
03|marketing|6521
04|personnel|2365
05|production|9876
06|sales|1006
adhoc@adhoc:~/Desktop$

Uniq option

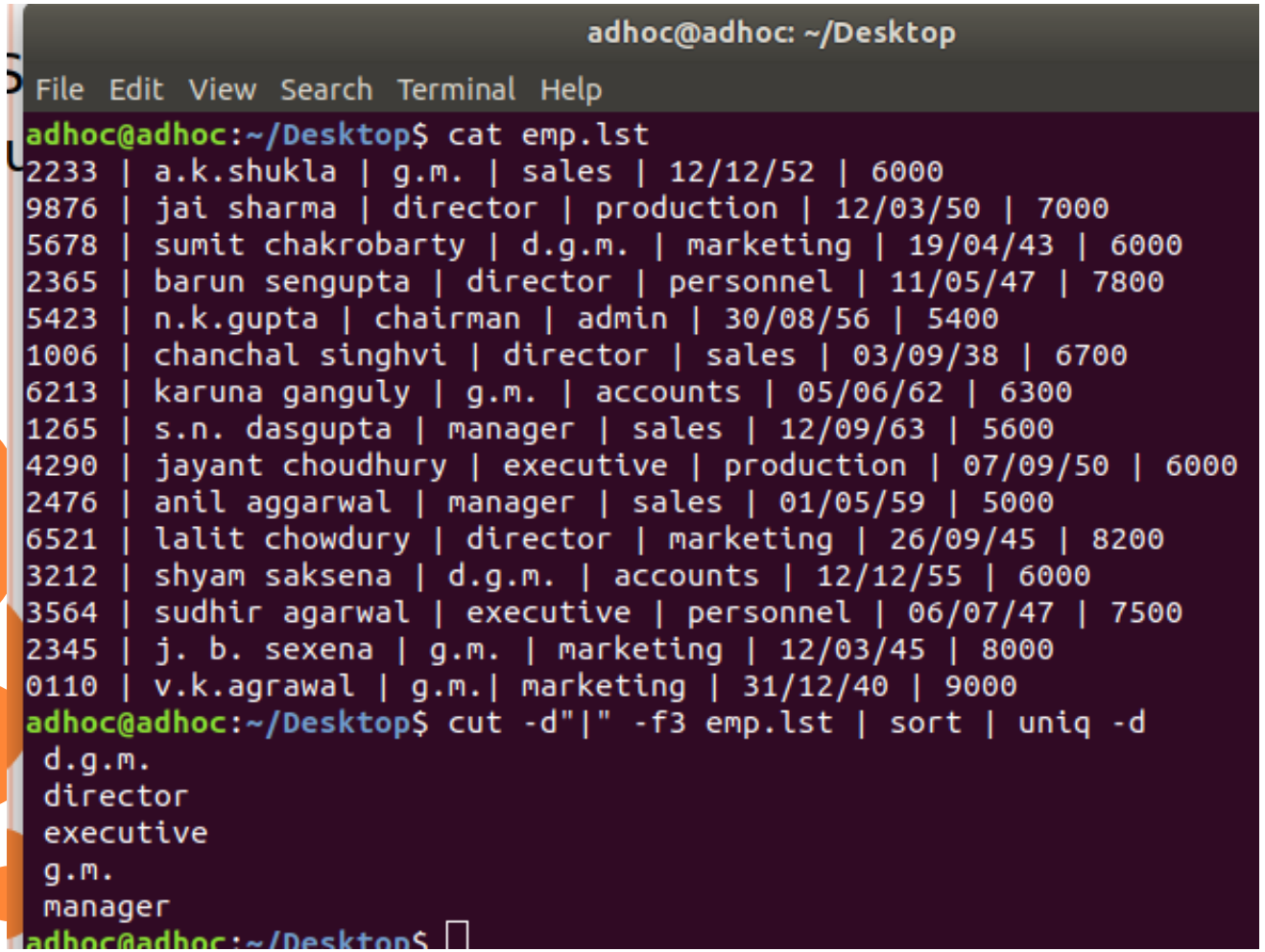
1. select nonrepeated lines (-u)

$cut -d“|” -f3 emp.lst | sort | uniq -u



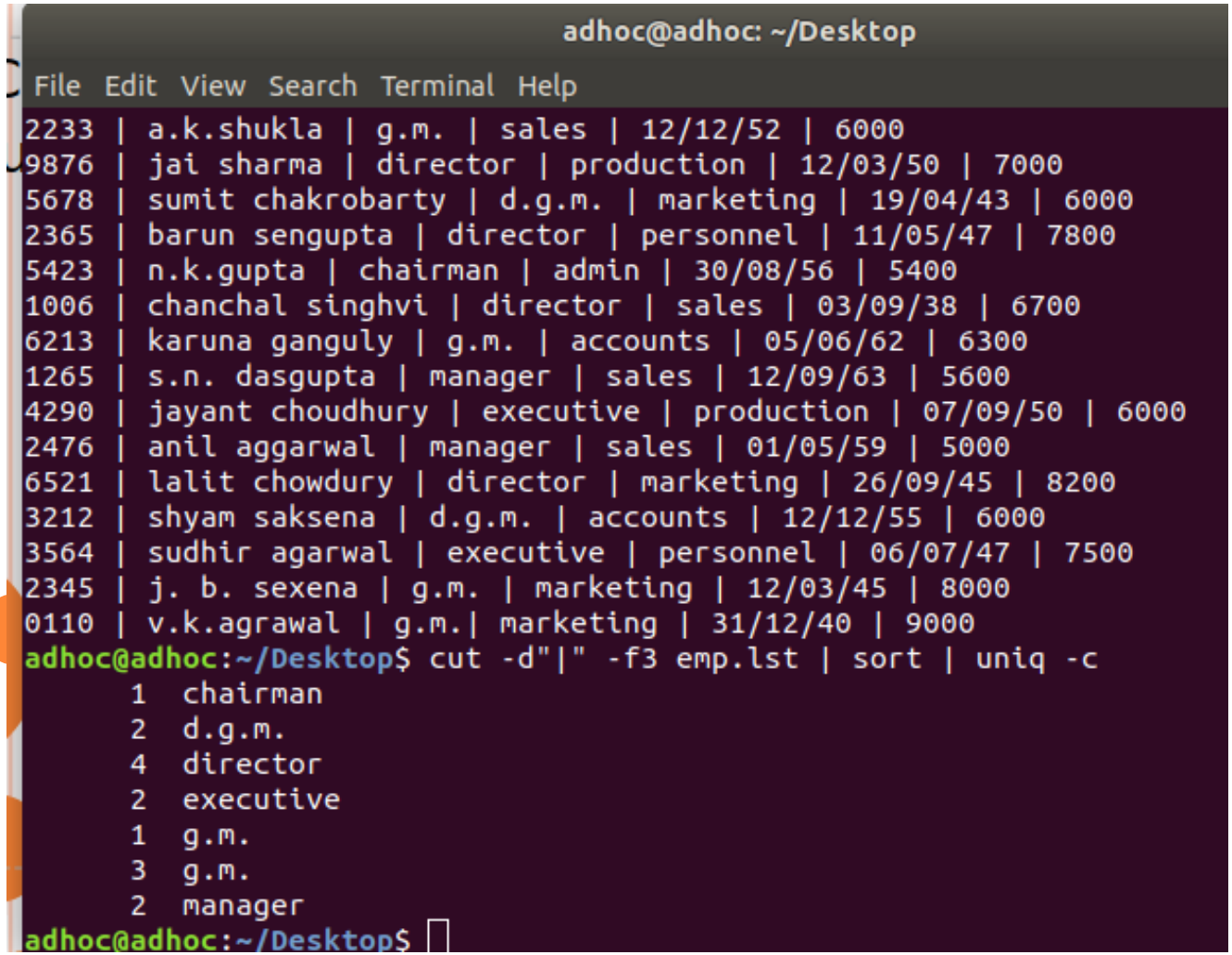
2.Select the duplicate line (-d)

$cut -d"|" -f3 emp.lst | sort |uniq -d



3.Couting the frquency of occurrence (-c)

$cut -d"|" -f3 emp.lst | sort |uniq -c

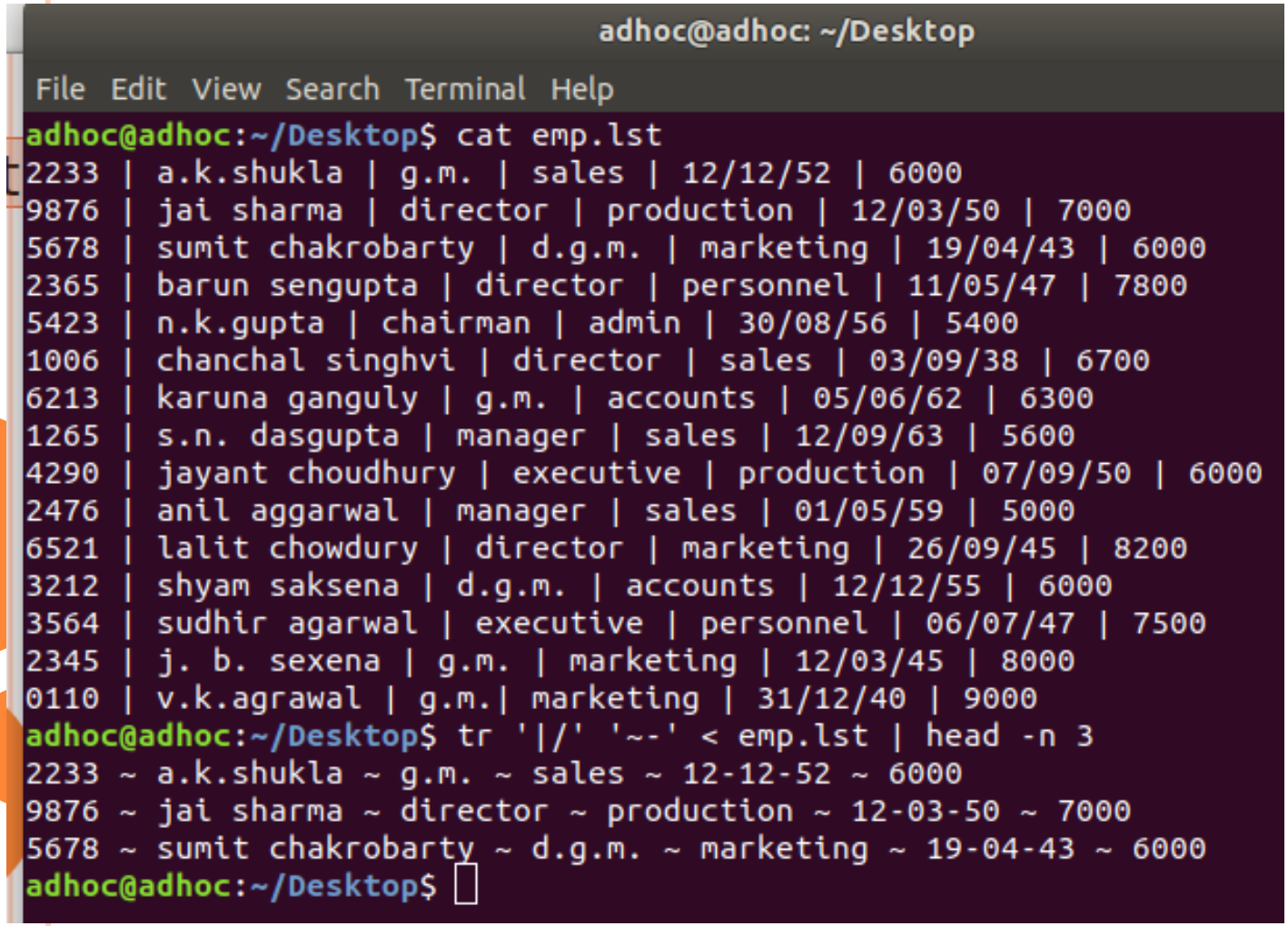


tr: Translating characters

- `tr` translates characters
- **Syntax :**
- tr options **exp1 exp2** standard input
  - Two arguments, given as strings
  - Default it translates each character in exp1 to its mapped counterpart in exp2
  - 1st char in exp1 is replace with the 1st char of exp2

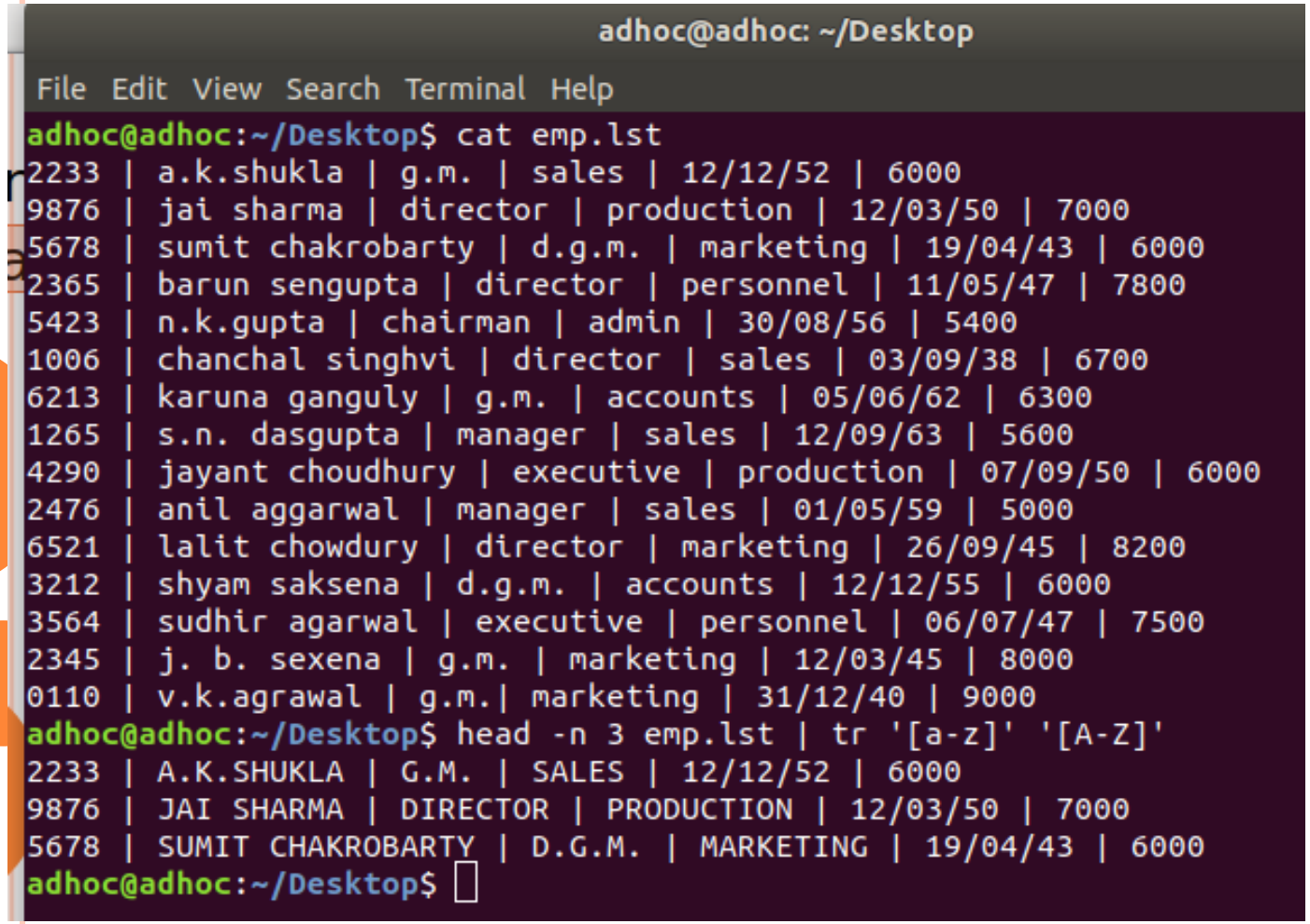
  Three options, c d s
  - Often used for letter case conversion
  - Useful for “cleaning up” formatted output

$tr '|/' '~-' < emp.lst | head -n 3



Changing the Case

- $head -n 3 emp.lst | tr '[a- z]' '[A- Z]'

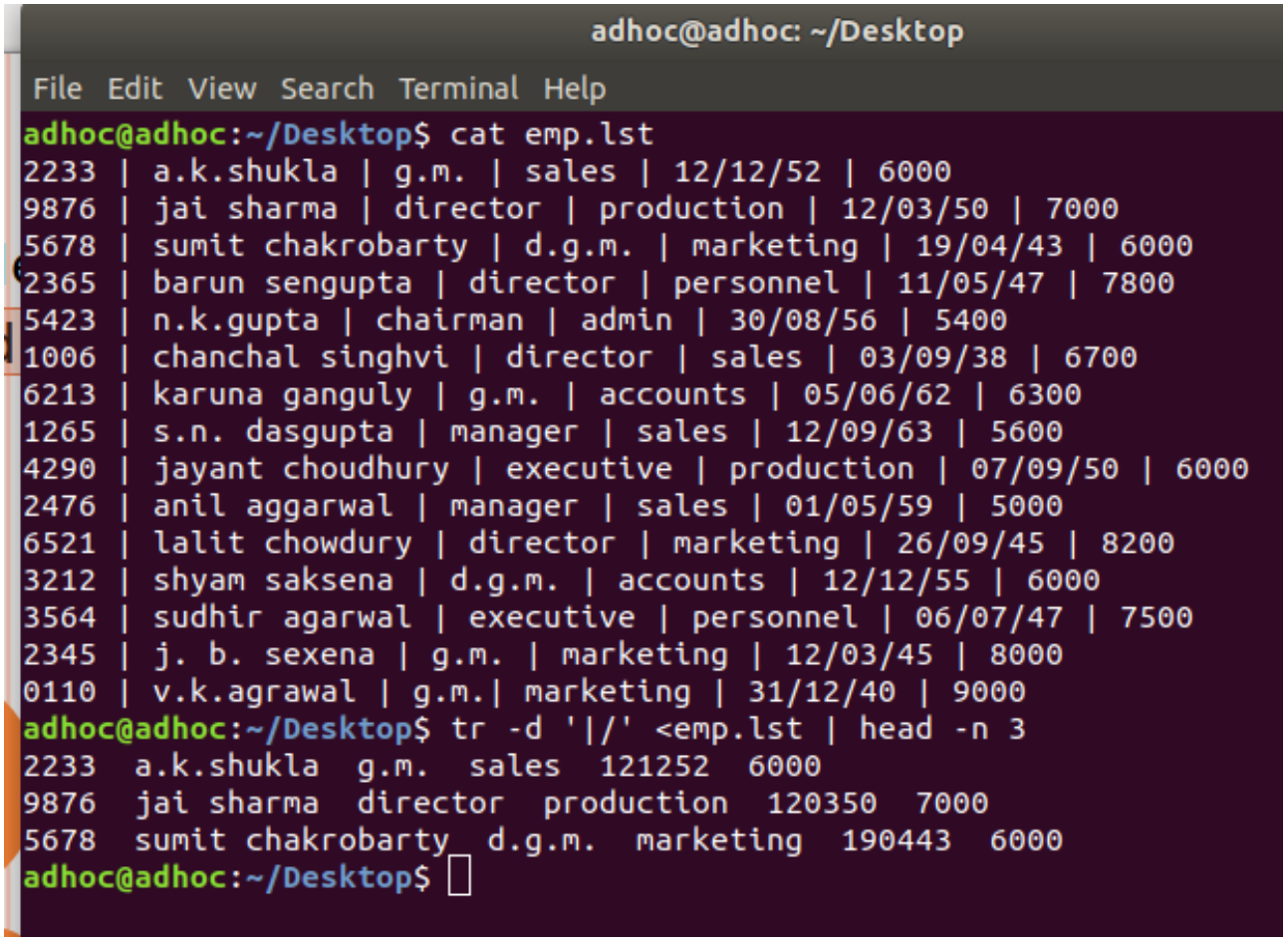


Tr option

1.Deleting character(-d)

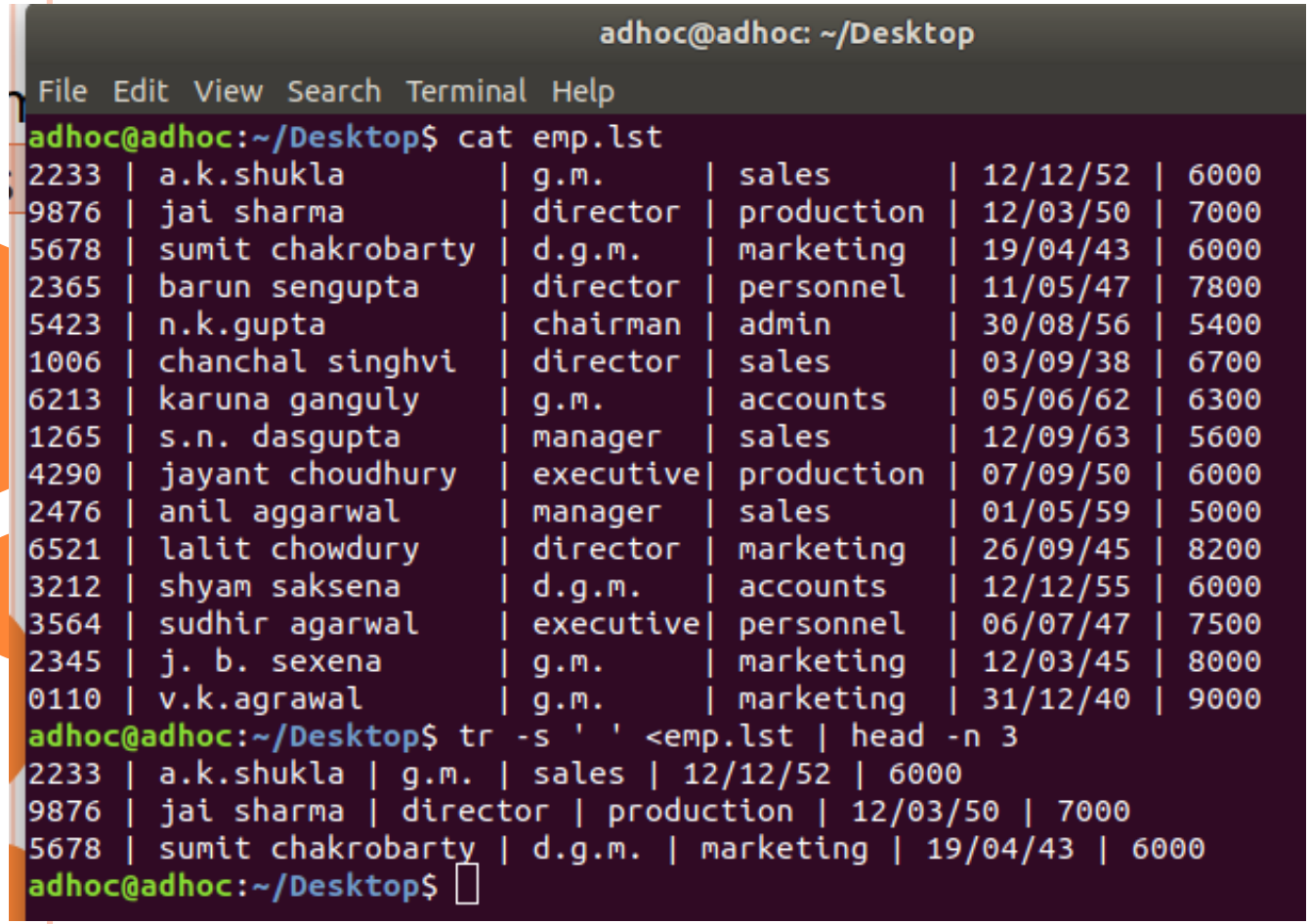
$tr -d '|/' <emp.lst | head -n 3

[delete | and /]



2.Compressing multiple consecutive characters(-s)

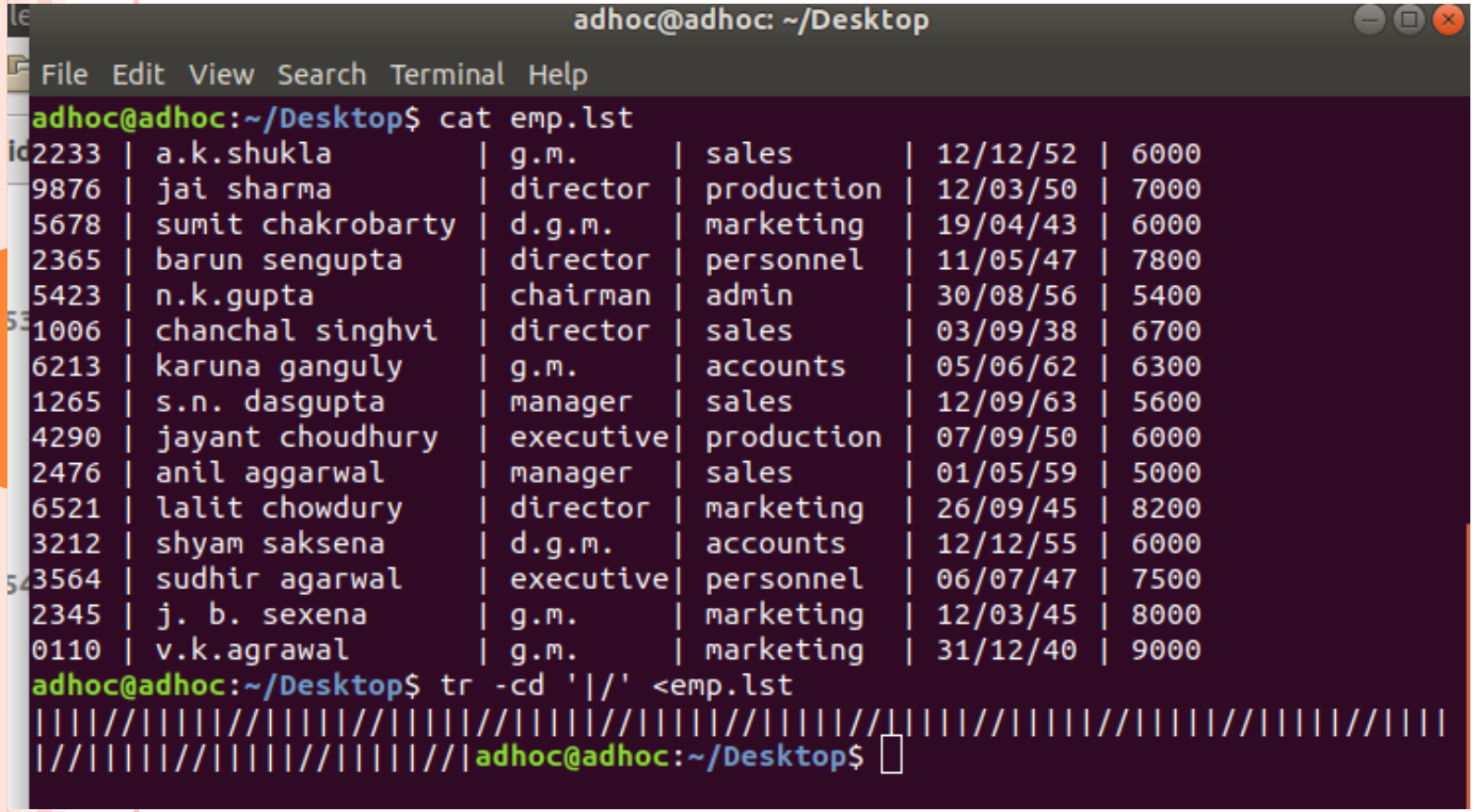
- $tr -s ‘ ‘ <emp.lst | head -n 3
- [squeezes multiple consecutive occurences of its agrument to a singal character]



2.complementing values of expression(-c)

$tr -cd ‘|/’ <emp.lst

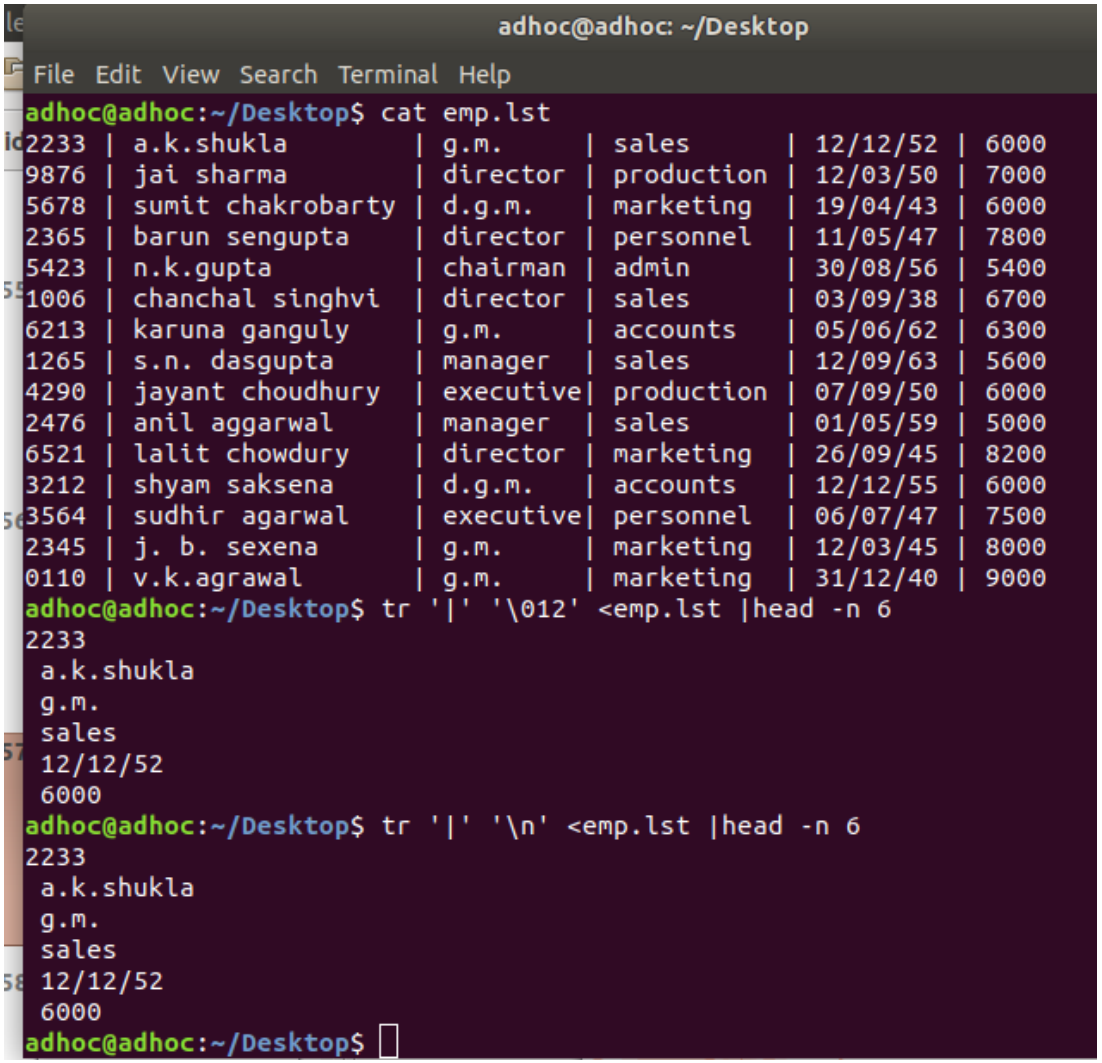
[delete all except |and \]


adhoc@adhoc: ~/Desktop
File Edit View Search Terminal Help
adhoc@adhoc:~/Desktop$ cat emp.lst
2233 | a.k.shukla | g.m. | sales | 12/12/52 | 6000
9876 | jai sharma | director | production | 12/03/50 | 7000
5678 | sumit chakrobarty | d.g.m. | marketing | 19/04/43 | 6000
2365 | barun sengupta | director | personnel | 11/05/47 | 7800
5423 | n.k.gupta | chairman | admin | 30/08/56 | 5400
1006 | chanchal singhvi | director | sales | 03/09/38 | 6700
6213 | karuna ganguly | g.m. | accounts | 05/06/62 | 6300
1265 | s.n. dasgupta | manager | sales | 12/09/63 | 5600
4290 | jayant choudhury | executive | production | 07/09/50 | 6000
2476 | anil aggarwal | manager | sales | 01/05/59 | 5000
6521 | lalit chowdury | director | marketing | 26/09/45 | 8200
3212 | shyam saksena | d.g.m. | accounts | 12/12/55 | 6000
3564 | sudhir agarwal | executive | personnel | 06/07/47 | 7500
2345 | j. b. sexena | g.m. | marketing | 12/03/45 | 8000
0110 | v.k.agrawal | g.m. | marketing | 31/12/40 | 9000
adhoc@adhoc:~/Desktop$ tr -cd '|/' <emp.lst
||||/|||||/|||||/|

3.Using ASCII octal values and Escape Sequance

$tr '|' '\012' <emp.lst |head -n 6

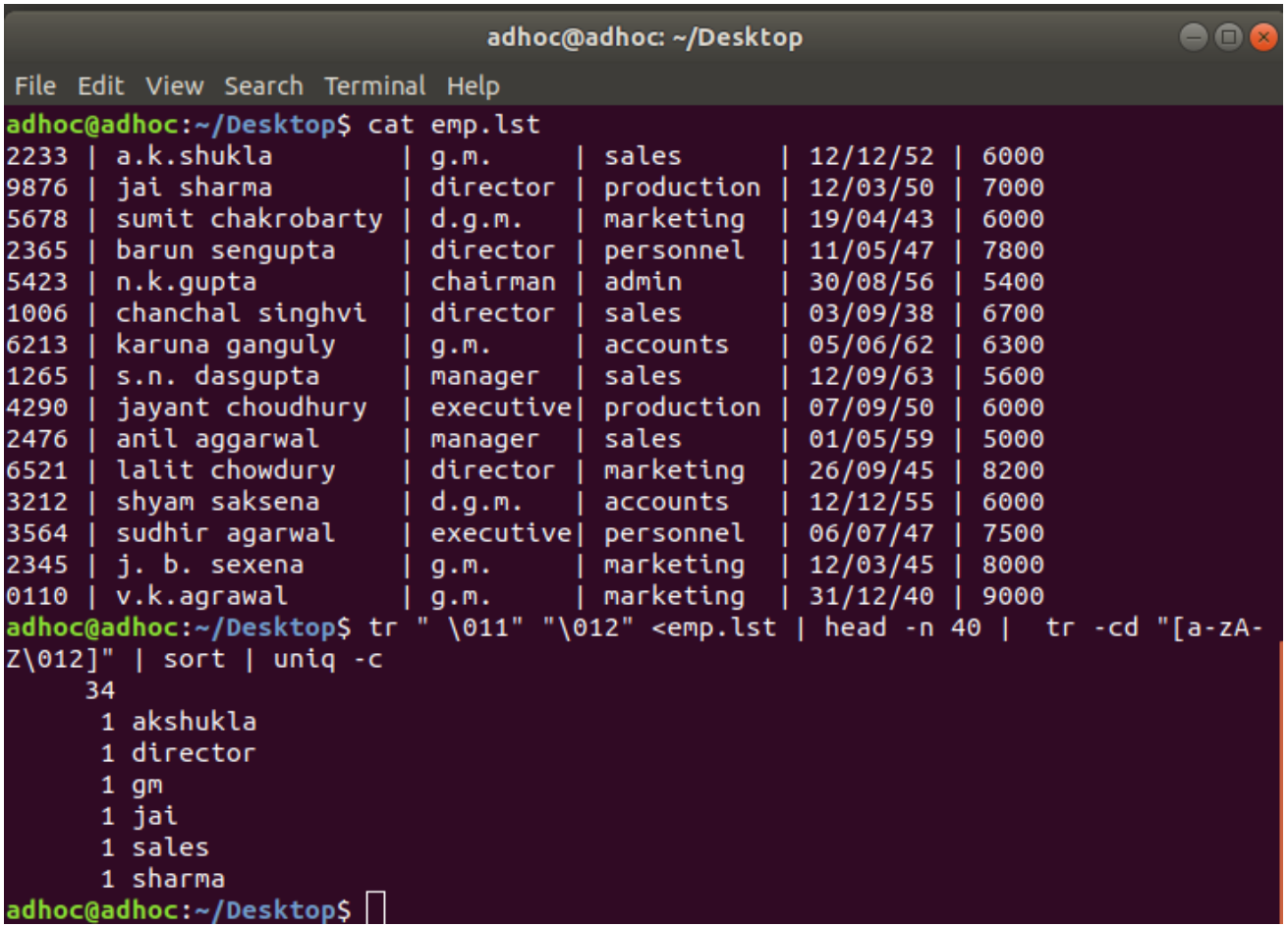
[replace pipe (|) with \012 or \n]



Displaying Word count

(-c):complementry (-d): delete

$tr " \011" "\012" <emp.lst | head -n 40 | tr -cd "[a-zA-Z\012]" | sort | uniq -c



Completed

Redirection I/O

Filter Defination

About Simple Filters

Filters – definition

To format text – **pr**

Pick lines from the beginning – **head**

Pick lines from the end – **tail**

Extract characters – **cut**

Join two lines / files – **paste**

Sort, merge and remove – **sort**

Find unique and no unique lines – **uniq**

Change, delete or squeeze characters - **tr**