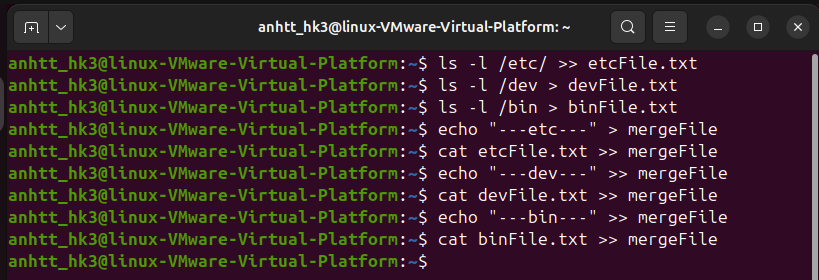
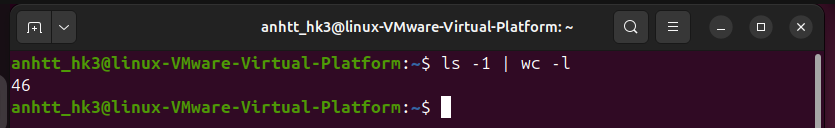
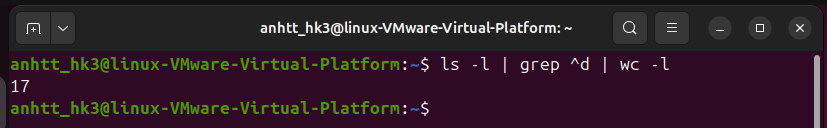
Hãy sử dụng các cơ chế định hướng lại vào/ra và pipe của Linux để thực hiện các nhiệm vụ sau:

1 Tạo ra ba *tệp* có nội dung là tên và thuộc tính của *các thư mục và tệp có trong các thư mục: /etc, /dev, /bin.* Ghép 3 tệp làm một, có phân vùng mỗi tệp.

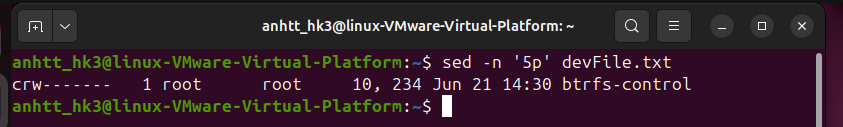
2 Đếm số lượng tệp và thư mục trong một thư mục (Vi du: ~)

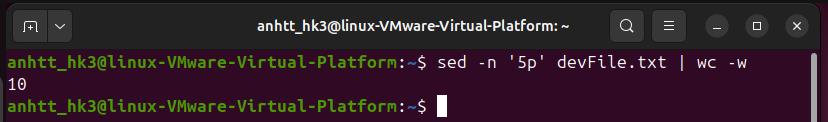


3 Đếm số lượng thư mục con của một thư mục (Vi du: ~)

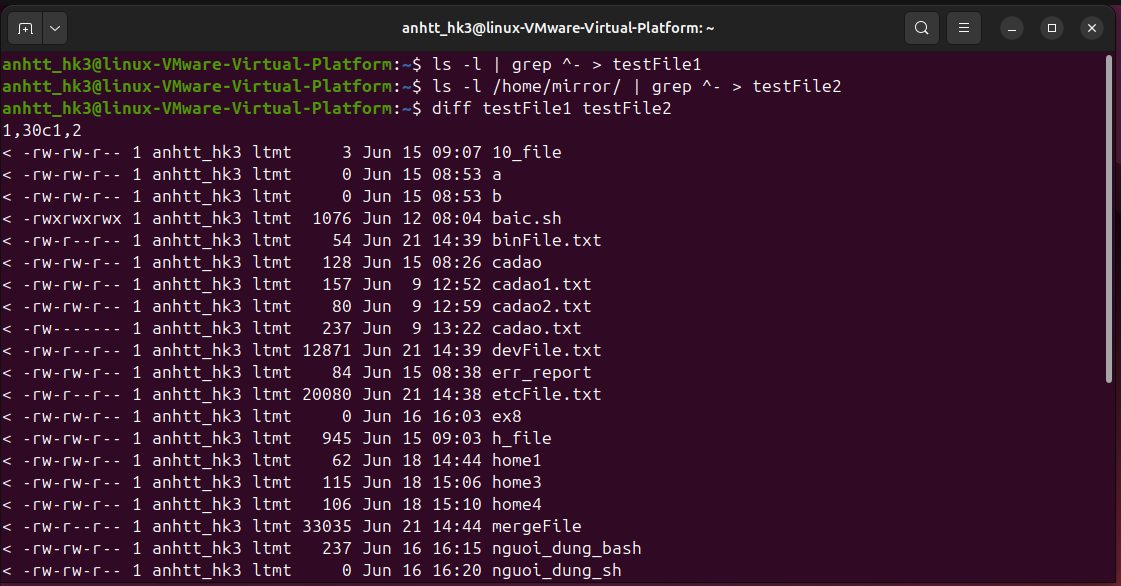


4 Cho một tệp văn bản, hãy in ra dòng thứ n bất kỳ của tệp và đếm số lượng từ trong dòng này

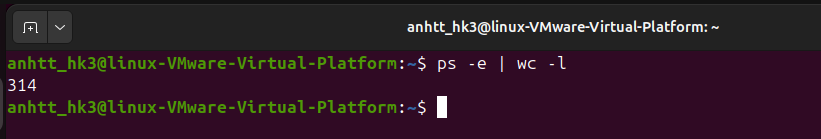
+ in ra dòng thứ 5

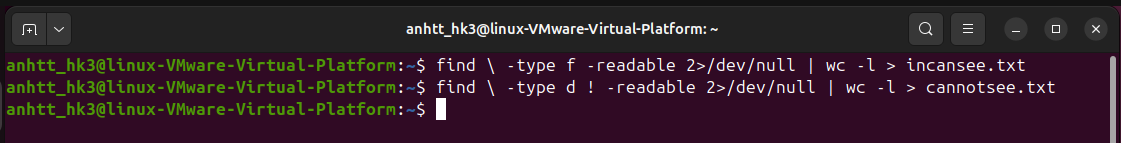
+ đếm số từ

6 Liệt kê sự khác nhau về tên tệp trong hai thư mục bất kỳ (sử dụng lệnh **diff**)

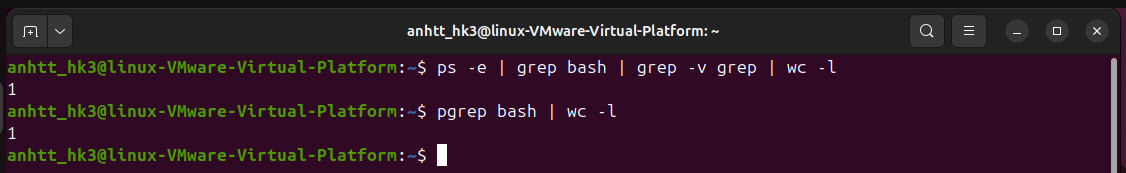
+ so sánh 2 tệp **~** và **/home/mirror**

7 Hãy đếm tổng số tiến trình đang có trong hệ thống

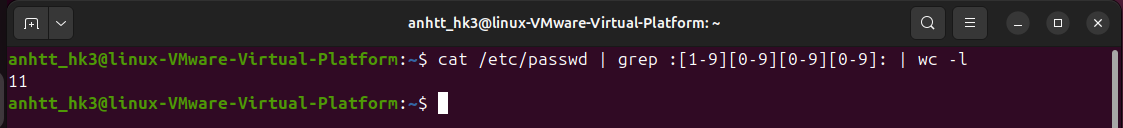


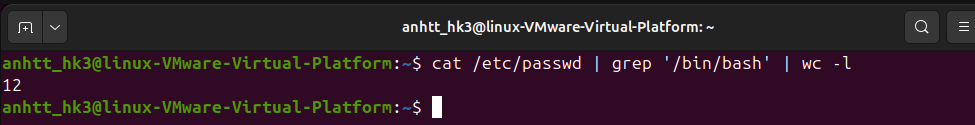
8 In ra số tệp tin trong hệ thống mà em có thể xem được vào trong file icansee.txt trong thư mục cá nhân. In ra các thư mục mà em không truy cập được vào file cannotsee.txt trong thư mục cá nhân.

9 Đếm số tiến trình được sinh ra từ bash trong hệ thống.

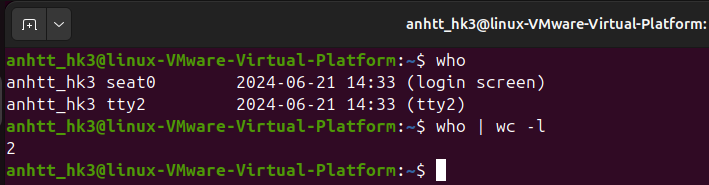


10 Đếm số lượng người sử dụng đã đăng ký với hệ thống (trong tệp /etc/passwd), và đếm số lượng người sử dụng đang sử dụng thông dịch lệnh **/bin/bash**

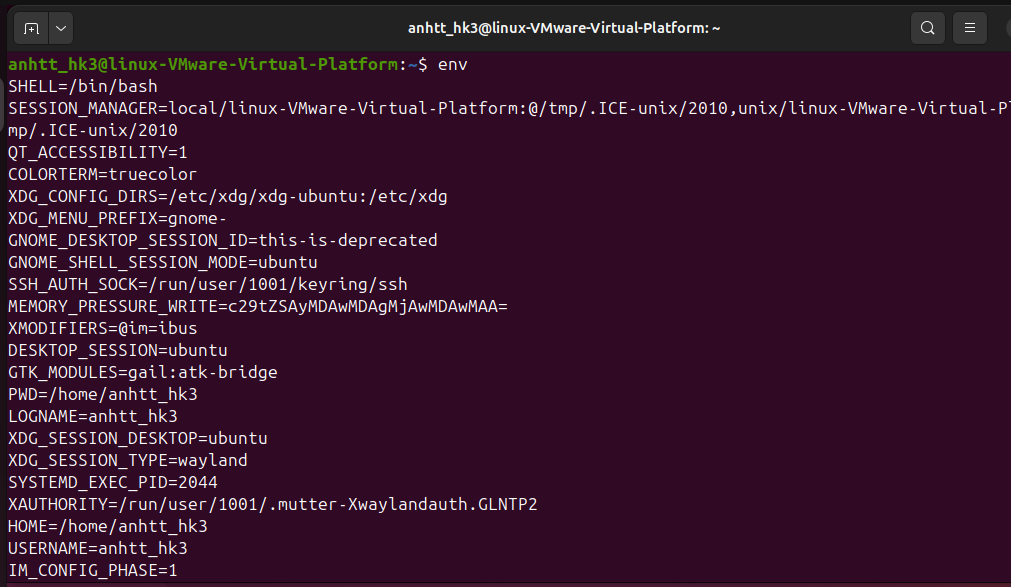
+ đến số người dùng đăng kí với hệ thống

+ số người đang sử dụng thông dịch /bin/bash:

11 Đếm số lượng người sử dụng đang đăng nhập vào hệ thống



12 Sử dụng lệnh **env** để xem giá trị các biến môi trường



Thực hiện theo các hướng dẫn sau để tìm hiểu các lệnh sed, cut, sort

Sed Command in Unix

SED command in UNIX is stands for stream editor and it can perform lot’s of function on file like, searching, find and replace, insertion or deletion. Though most common use of SED command in UNIX is for substitution or for find and replace. By using SED you can edit files even without opening it, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it.

* SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution).
* SED command in unix supports regular expression which allows it perform complex pattern matching.

**Syntax:**

**sed OPTIONS... [SCRIPT] [INPUTFILE...]**

**Example:**  
Consider the below text file as an input.

**$cat > geekfile.txt**

unix is great os. unix is opensource. unix is free os.

learn operating system.

unix linux which one you choose.

unix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

**Sample Commands**

1. **Replacing or substituting string :** Sed command is mostly used to replace the text in a file. The below simple sed command replaces the word “unix” with “linux” in the file.
2. **$sed 's/unix/linux/' geekfile.txt**

**Output :**

linux is great os. unix is opensource. unix is free os.

learn operating system.

linux linux which one you choose.

linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

Here the “s” specifies the substitution operation. The “/” are delimiters. The “unix” is the search pattern and the “linux” is the replacement string.

By default, the sed command replaces the first occurrence of the pattern in each line and it won’t replace the second, third…occurrence in the line.

1. **Replacing the nth occurrence of a pattern in a line :**Use the /1, /2 etc flags to replace the first, second occurrence of a pattern in a line. The below command replaces the second occurrence of the word “unix” with “linux” in a line.
2. **$sed 's/unix/linux/2' geekfile.txt**

**Output:**

unix is great os. linux is opensource. unix is free os.

learn operating system.

unix linux which one you choose.

unix is easy to learn.linux is a multiuser os.Learn unix .unix is a powerful.

1. **Replacing all the occurrence of the pattern in a line :**The substitute flag /g (global replacement) specifies the sed command to replace all the occurrences of the string in the line.
2. **$sed 's/unix/linux/g' geekfile.txt**

**Output :**

linux is great os. linux is opensource. linux is free os.

learn operating system.

linux linux which one you choose.

linux is easy to learn.linux is a multiuser os.Learn linux .linux is a powerful.

1. **Replacing from nth occurrence to all occurrences in a line :**Use the combination of /1, /2 etc and /g to replace all the patterns from the nth occurrence of a pattern in a line. The following sed command replaces the third, fourth, fifth… “unix” word with “linux” word in a line.
2. **$sed 's/unix/linux/3g' geekfile.txt**

**Output:**

unix is great os. unix is opensource. linux is free os.

learn operating system.

unix linux which one you choose.

unix is easy to learn.unix is a multiuser os.Learn linux .linux is a powerful.

1. **Parenthesize first character of each word :**This sed example prints the first character of every word in paranthesis.
2. **$ echo "Welcome To The Geek Stuff" | sed 's/\(\b[A-Z]\)/\(\1\)/g'**

Output:

(W)elcome (T)o (T)he (G)eek (S)tuff

1. **Replacing string on a specific line number :**You can restrict the sed command to replace the string on a specific line number. An example is
2. **$sed '3 s/unix/linux/' geekfile.txt**

**Output:**

unix is great os. unix is opensource. unix is free os.

learn operating system.

linux linux which one you choose.

unix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

The above sed command replaces the string only on the third line.

1. **Duplicating the replaced line with /p flag :**The /p print flag prints the replaced line twice on the terminal. If a line does not have the search pattern and is not replaced, then the /p prints that line only once.
2. **$sed 's/unix/linux/p' geekfile.txt**

**Output:**

linux is great os. unix is opensource. unix is free os.

linux is great os. unix is opensource. unix is free os.

learn operating system.

linux linux which one you choose.

linux linux which one you choose.

linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

1. **Printing only the replaced lines :**Use the -n option along with the /p print flag to display only the replaced lines. Here the -n option suppresses the duplicate rows generated by the /p flag and prints the replaced lines only one time.
2. **$sed -n 's/unix/linux/p' geekfile.txt**

**Output:**

linux is great os. unix is opensource. unix is free os.

linux linux which one you choose.

linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

If you use -n alone without /p, then the sed does not print anything.

1. **Replacing string on a range of lines :**You can specify a range of line numbers to the sed command for replacing a string.
2. **$sed '1,3 s/unix/linux/' geekfile.txt**

**Output:**

linux is great os. unix is opensource. unix is free os.

learn operating system.

linux linux which one you choose.

unix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

Here the sed command replaces the lines with range from 1 to 3. Another example is

**$sed '2,$ s/unix/linux/' geekfile.txt**

**Output:**

unix is great os. unix is opensource. unix is free os.

learn operating system.

linux linux which one you choose.

linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful

Here $ indicates the last line in the file. So the sed command replaces the text from second line to last line in the file.

This article is contributed by **Akshay Rajput**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](http://www.contribute.geeksforgeeks.org/) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

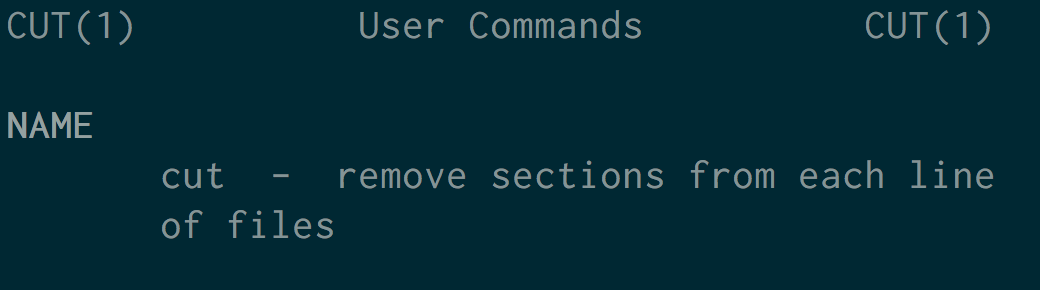
# *Linux and Unix cut command tutorial with examples*

## Tutorial on using cut, a UNIX and Linux command for cutting sections from each line of files. Examples of cutting by character, byte position, cutting based on delimiter and how to modify the output delimiter.

*Estimated reading time: 3 minutes*

## Table of contents

* + [What is the cut command in UNIX?](https://shapeshed.com/unix-cut/#what-is-the-cut-command-in-unix)
  + [How to cut by byte position](https://shapeshed.com/unix-cut/#how-to-cut-by-byte-position)
  + [How to cut by character](https://shapeshed.com/unix-cut/#how-to-cut-by-character)
  + [How to cut based on a delimiter](https://shapeshed.com/unix-cut/#how-to-cut-based-on-a-delimiter)
  + [How to cut by complement pattern](https://shapeshed.com/unix-cut/#how-to-cut-by-complement-pattern)
  + [How to modify the output delimiter](https://shapeshed.com/unix-cut/#how-to-modify-the-output-delimiter)
  + [Further reading](https://shapeshed.com/unix-cut/#further-reading)



## What is the cut command in UNIX?

The cut command in UNIX is a command line utility for cutting sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character and delimiter. It can also be used to cut data from file formats like CSV.

## How to cut by byte position

To cut out a section of a line by specifying a byte position use the -b option.

echo 'baz' | cut -b 2

a

echo 'baz' | cut -b 1-2

ba

echo 'baz' | cut -b 1,3

bz

## How to cut by character

To cut by character use the -c option. This selects the characters given to the -c option. This can be a list of comma separated numbers, a range of numbers or a single number.

Where your input stream is character based -c can be a better option than selecting by bytes as often characters are more than one byte.

In the following example character ‘♣’ is three bytes. By using the -c option the character can be correctly selected along with any other characters that are of interest.

echo '♣foobar' | cut -c 1,6

♣a

echo '♣foobar' | cut -c 1-3

♣fo

## How to cut based on a delimiter

To cut using a delimiter use the -d option. This is normally used in conjunction with the -f option to specify the field that should be cut.

In the following example a CSV file exists and is saved as names.csv.

John,Smith,34,London

Arthur,Evans,21,Newport

George,Jones,32,Truro

The delimiter can be set to a comma with -d ','. cut can then pull out the fields of interest with the -f flag. In the following example the first field is cut.

cut -d ',' -f 1 names.csv

John

Arthur

George

Multiple fields can be cut by passing a comma separated list.

cut -d ',' -f 1,4 names.csv

John,London

Arthur,Newport

George,Truro

## How to cut by complement pattern

To cut by complement us the --complement option. Note this option is not available on the BSD version of cut. The --complement option selects the inverse of the options passed to sort.

In the following example the -c option is used to select the first character. Because the --complement option is also passed to cut the second and third characters are cut.

echo 'foo' | cut --complement -c 1

oo

## How to modify the output delimiter

To modify the output delimiter use the --output-delimiter option. Note that this option is not available on the BSD version of cut. In the following example a semi-colon is converted to a space and the first, third and fourth fields are selected.

echo 'how;now;brown;cow' | cut -d ';' -f 1,3,4 --output-delimiter=' '

how brown cow

15 examples of sort command in Linux

sort command is used to sort a file, arranging the records in a particular order. By default, the sort command sorts file assuming the contents are ascii. Using options in sort command, it can also be used to sort numerically. Let us discuss it with some examples:

File with Ascii data:

Let us consider a file with the following contents:

$ cat file

Unix

Linux

Solaris

AIX

Linux

HPUX

1. sort simply sorts the file in alphabetical order:

$ sort file

AIX

HPUX

Linux

Linux

Solaris

Unix

All records are sorted alphabetically.

2. sort removes the duplicates using the -u option:

$ sort -u file

AIX

HPUX

Linux

Solaris

Unix

The duplicate 'Linux' record got removed. '-u' option removes all the duplicate records in the file. Even if the file have had 10 'Linux' records, with -u option, only the first record is retained.

File with numbers:

Let us consider a file with numbers:

$ cat file

20

19

5

49

200

3. The default sort 'might' give incorrect result on a file containing numbers:

$ sort file

19

20

200

49

5

In the above result, 200 got placed immediately below 20, not at the end which is incorrect. This is because the sort did ASCII sort. If the file had not contained '200', the default sort would have given proper result. However, it is incorrect to sort a numerical file in this way since the sorting logic is incorrect.

4. To sort a file numericallly:

$ sort -n file

5

19

20

49

200

-n option can sort the decimal numbers as well.

5. sort file numerically in reverse order:

$ sort -nr file

200

49

20

19

5

'r' option does a reverse sort.

Multiple Files:

Let us consider examples with multiple files, say file1 and file2, containing numbers:

$ cat file1

20

19

5

49

200

$ cat file2

25

18

5

48

200

6. sort can sort multiple files as well.

$ sort -n file1 file2

5

5

18

19

20

25

48

49

200

200

The result of sort with multiple files will be a sorted and merged output of the multiple files.

7. Sort, merge and remove duplicates:

$ sort -nu file1 file2

5

18

19

20

25

48

49

200

-u option becomes more handy in case of multiple files. With this, the output is now sorted, merged and without duplicate records.

Files with multiple fields and delimiter:

Let us consider a file with multiple fields:

$ cat file

Linux,20

Unix,30

AIX,25

Linux,25

Solaris,10

HPUX,100

8. sorting a file containing multiple fields:

$ sort file

AIX,25

HPUX,100

Linux,20

Linux,25

Solaris,10

Unix,30

As shown above, the file got sorted on the 1st field, by default.

9. sort file on the basis of 1st field:

$ sort -t"," -k1,1 file

AIX,25

HPUX,100

Linux,20

Linux,25

Solaris,10

Unix,30

This is being more explicit. '-t' option is used to provide the delimiter in case of files with delimiter. '-k' is used to specify the keys on the basis of which the sorting has to be done. The format of '-k' is : '-km,n' where m is the starting key and n is the ending key. In other words, sort can be used to sort on a range of fields just like how the group by in sql does. In our case, since the sorting is on the 1st field alone, we speciy '1,1'. Similarly, if the sorting is to be done on the basis of first 3 fields, it will be: '-k 1,3'.

Note: For a file which has fields delimited by a space or a tab, there is no need to specify the "-t" option since the white space is the delimiter by default in sort.

10. sorting file on the basis of the 2nd field:

$ sort -t"," -k2,2 file

Solaris,10

HPUX,100

Linux,20

AIX,25

Linux,25

Unix,30

11. sorting file on the basis of 2nd field , numerically:

$ sort -t"," -k2n,2 file

Solaris,10

Linux,20

AIX,25

Linux,25

Unix,30

HPUX,100

12. Remove duplicates from the file based on 1st field:

$ sort -t"," -k1,1 -u file

AIX,25

HPUX,100

Linux,20

Solaris,10

Unix,30

The duplicate Linux record got removed. Keep in mind, the command "sort -u file" would not have worked here becuase both the 'Linux' records are not same, the values were different. However, in the above, sort is told to remove the duplicates based on the 1st key, and hence the duplicate 'Linux' record got removed. According to sort, in case of a group of similar records, except the first one, the rest are considered duplicate.

13. Sort the file numerically on the 2nd field in reverse order:

$ sort -t"," -k2nr,2 file

HPUX,100

Unix,30

AIX,25

Linux,25

Linux,20

Solaris,10

14. sort the file alphabetically on the 1st field, numerically on the 2nd field:

$ sort -t"," -k1,1 -k2n,2 file

AIX,25

HPUX,100

Linux,20

Linux,25

Solaris,10

Unix,30

15. sort a file based on the 1st and 2nd field, and numerically on 3rd field on a file containing 5 columns:

$ sort -t"," -k1,2 -k3n,3 file