

## Join Command in Unix/Linux Examples

Join command is one of the text processing utility in Unix/Linux. Join command is used to combine two files based on a matching fields in the files. If you know SQL, the join command is similar to joining two tables in a database.

The syntax of join command is

```
join [options] file1 file2
```

The join command options are

```
-1 field number : Join on the specified field number in the first file
-2 field number : Join on the specified field number in the second file
-j field number : Equivalent to -1 fieldnumber and -2 fieldnumber
-o list : displays only the specified fields from both the files
-t char : input and output field delimiter
-a filename : Prints non matched lines in a file
-i : ignore case while joining
```

### Popular Posts

[Top Examples of Awk Command in Unix](#)[Sed Command in Unix and Linux Examples](#)[Informatica Scenario Based Interview Questions with Answers - Part 1](#)[Cut Command in Unix \( Linux\) Examples](#)[SQL Queries Interview Questions - Oracle Part 1](#)[Find Command in Unix and Linux Examples](#)[Date Functions in Hive](#)[Informatica Scenario Based Questions - Part 2](#)[String Functions in Hive](#)[Unix Sed Command to Delete Lines in File - 15 Examples](#)

### Have Questions? Follow Me

## Unix Join Command Examples


1. Write a join command to join two files on the first field?

The basic usage of join command is to join two files on the first field. By default the join command matches the files on the first fields when we do not specify the field numbers explicitly. Let's say we have two files emp.txt and dept.txt

```
> cat emp.txt
10 mark
10 steve
20 scott
30 chris
> cat dept.txt
10 hr
20 finance
30 db
```

Here we will join on the first field and see the output. By default, the join command treats the field delimiter as space or tab.

```
> join emp.txt dept.txt
10 mark hr
10 steve hr
20 scott finance
30 chris db
```

 vijay bhaskar

 Add to circles



267 have me in circles

[View all](#)

Important Note: Before joining the files, make sure to sort the fields on the joining fields. Otherwise you will get incorrect result.

2. Write a join command to join the two files? Here use the second field from the first file and the first field from the second file to join.

In this example, we will see how to join two files on different fields rather than the first field. For this consider the below two files as an example

```
> cat emp.txt
mark 10 1
steve 10 1
scott 20 2
chris 30 3
> cat dept.txt
10 hr 1
20 finance 2
30 db 3
```

From the above, you can see the join fields are the second field from the emp.txt and the first field from the dept.txt. The join command to match these two files is

```
> join -1 2 -2 1 emp.txt dept.txt
10 mark 1 hr 1
10 steve 1 hr 1
20 scott 2 finance 2
30 chris 3 db 3
```

You can also see that the two files can also be joined on the third field. As both the files have the matching join field, you can use the `j` option in the join command.

Here `-1 2` specifies the second field from the first file (`emp.txt`) and `-2 1` specifies the first field from the second file (`dept.txt`)

```
> join -j 3 emp.txt dept.txt
1 mark 10 10 hr
1 steve 10 10 hr
2 scott 20 20 finance
3 chris 30 30 db
```

**3.** Write a join command to select the required fields from the input files in the output? Select first field from first file and second field from second file in the output.

By default, the join command prints all the fields from both the files (except the join field is printed once). We can choose what fields to be printed on the terminal with the `-o` option. We will use the same files from the above example.

```
> join -o 1.1 2.2 -1 2 -2 1 emp.txt dept.txt
mark hr
steve hr
scott finance
chris db
```

Here `1.1` means in the first file select the first field. Similarly, `2.2` means in the second file

select the second field

4. Write a command to join two delimited files? Here the delimiter is colon (:)

So far we have joined files with space delimiter. Here we will see how to join files with a colon as delimiter. Consider the below two files.

```
> cat emp.txt
mark:10
steve:10
scott:20
chris:30
> cat dept.txt
10:hr
20:finance
30:db
```

The -t option is used to specify the delimiter. The join command for joining the files is

```
> join -t: -1 2 -2 1 emp.txt dept.txt
10:mark:hr
10:steve:hr
20:scott:finance
30:chris:db
```

5. Write a command to ignore case when joining the files?

If the join fields are in different cases, then the join will not be performed properly. To ignore the case in join use the -i option.

```
> cat emp.txt
mark,A
steve,a
scott,b
chris,C
> cat dept.txt
a,hr
B,finance
c,db

> join -t, -i -1 2 -2 1 emp.txt dept.txt
A,mark,hr
a,steve,hr
b,scott,finance
C,chris,db
```

6. Write a join command to print the lines which do not match the values in joining fields?

By default the join command prints only the matched lines from both the files which means prints the matched lines that passed the join condition. We can use the -a option to print the non-matched lines.

```
> cat P.txt
A 1
B 2
C 3
```

```
> cat Q.txt
```

```
B 2
```

```
C 3
```

```
D 4
```

Print non pairable lines from first file.

```
> join -a 1 P.txt Q.txt
```

```
A 1
```

```
B 2 2
```

```
C 3 3
```

Print non pairable lines from second file.

```
> join -a 2 P.txt Q.txt
```

```
B 2 2
```

```
C 3 3
```

```
D 4
```

Print non pairable lines from both file.

```
> join -a 1 -a 2 P.txt Q.txt
```

```
A 1
```

```
B 2 2
```

```
C 3 3
```

```
D 4
```