NAME

join - relational database operator

SYNOPSIS

join [**-a** *file_number* | **-v** *file_number*] [**-e** *string*] [**-o** *list*] [**-t** *char*] [**-1** *field*] [**-2** *field*] *file1 file2*

DESCRIPTION

The **join** utility performs an "equality join" on the specified files and writes the result to the standard output. The "join field" is the field in each file by which the files are compared. The first field in each line is used by default. There is one line in the output for each pair of lines in *file1* and *file2* which have identical join fields. Each output line consists of the join field, the remaining fields from *file1* and then the remaining fields from *file2*.

The default field separators are tab and space characters. In this case, multiple tabs and spaces count as a single field separator, and leading tabs and spaces are ignored. The default output field separator is a single space character.

Many of the options use file and field numbers. Both file numbers and field numbers are 1 based, i.e., the first file on the command line is file number 1 and the first field is field number 1. The following options are available:

-a *file_number*
: In addition to the default output, produce a line for each unpairable line in file *file_number*.

-e *string*
: Replace empty output fields with *string*.

-o *list*
: The **-o** option specifies the fields that will be output from each file for each line with matching join fields. Each element of *list* has either the form *file_number.field*, where *file_number* is a file number and *field* is a field number, or the form ‘0’ (zero), representing the join field. The elements of list must be either comma (‘,’) or whitespace separated. (The latter requires quoting to protect it from the shell, or, a simpler approach is to use multiple **-o** options.)

-t *char*
: Use character *char* as a field delimiter for both input and output. Every occurrence of *char* in a line is significant.

-v *file_number*
: Do not display the default output, but display a line for each unpairable line in file *file_number*. The options **-v 1** and **-v 2** may be specified at the same time.

-1 *field*
Join on the *field*'th field of *file1*.

-2 *field*
Join on the *field*'th field of *file2*.

When the default field delimiter characters are used, the files to be joined should be ordered in the collating sequence of sort(1), using the **-b** option, on the fields on which they are to be joined, otherwise **join** may not report all field matches. When the field delimiter characters are specified by the **-t** option, the collating sequence should be the same as sort(1) without the **-b** option.

If one of the arguments *file1* or *file2* is '**-**', the standard input is used.

EXIT STATUS

The **join** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

Assuming a file named *nobel_laureates.txt* with information about some of the first Nobel Peace Prize laureates:

```
1901,Jean Henri Dunant,M
1901,Frederic Passy,M
1902,Elie Ducommun,M
1905,Baroness Bertha Sophie Felicita Von Suttner,F
1910,Permanent International Peace Bureau,
```

and a second file *nobel_nationalities.txt* with their nationalities:

```
Jean Henri Dunant,Switzerland
Frederic Passy,France
Elie Ducommun,Switzerland
Baroness Bertha Sophie Felicita Von Suttner
```

Join the two files using the second column from first file and the default first column from second file specifying a custom field delimiter:

```
$ join -t, -1 2 nobel_laureates.txt nobel_nationalities.txt
Jean Henri Dunant,1901,M,Switzerland
Frederic Passy,1901,M,France
Elie Ducommun,1902,M,Switzerland
```

```
Baroness Bertha Sophie Felicita Von Suttner,1905,F
```

Show only the year and the nationality of the laureate using ‘<<NULL>>’ to replace empty fields:

```
$ join -e "<<NULL>>" -t, -1 2 -o "1.1 2.2" nobel_laureates.txt nobel_nationalities.txt
1901,Switzerland
1901,France
1902,Switzerland
1905,<<NULL>>
```

Show only lines from first file which do not have a match in second file:

```
$ join -v1 -t, -1 2 nobel_laureates.txt nobel_nationalities.txt
Permanent International Peace Bureau,1910,
```

Assuming a file named *capitals.txt* with the following content:

```
Belgium,Brussels
France,Paris
Italy,Rome
Switzerland
```

Show the name and capital of the country where the laureate was born. This example uses *nobel_nationalities.txt* as a bridge but does not show any information from that file. Also see the note about sort(1) above to understand why we need to sort the intermediate result.

```
$ join -t, -1 2 -o 1.2 2.2 nobel_laureates.txt nobel_nationalities.txt | \
   sort -k2 -t, | join -t, -e "<<NULL>>" -1 2 -o 1.1 2.2 - capitals.txt
Elie Ducommun,<<NULL>>
Jean Henri Dunant,<<NULL>>
```

COMPATIBILITY

For compatibility with historic versions of **join**, the following options are available:

-a In addition to the default output, produce a line for each unpairable line in both *file1* and *file2*.

-j1 *field*
Join on the *field*’th field of *file1*.

-j2 *field*

Join on the *field*’th field of *file2*.

-j *field*

Join on the *field*’th field of both *file1* and *file2*.

-o *list ...*

Historical implementations of **join** permitted multiple arguments to the **-o** option. These arguments were of the form *file_number*.*field_number* as described for the current **-o** option. This has obvious difficulties in the presence of files named *1.2*.

These options are available only so historic shell scripts do not require modification and should not be used.

SEE ALSO

awk(1), comm(1), paste(1), sort(1), uniq(1)

STANDARDS

The **join** command conforms to IEEE Std 1003.1-2001 ("POSIX.1").