**Лабораторная работа №20**

Итак, начнём. В данной лабораторной работе я расскажу о 23 утилитах bash. Из них 3 будут выделены и рассказаны более подробно, чем остальные. Приступим!

1. **cmp**

Команда cmp сравнивает два файла и, если они отличаются, сообщает о первом байте и строке, в которой найдено различие:

root@DESKTOP-5HM2HTK:~# cat file1

Hell

World

Less

root@DESKTOP-5HM2HTK:~# cat file2

Hell

Qrob

Frog

root@DESKTOP-5HM2HTK:~# cmp file1 file2

file1 file2 differ: byte 6, line 2

root@DESKTOP-5HM2HTK:~#

Если файлы одинаковые, то на вывод передано ничего не будет:

root@DESKTOP-5HM2HTK:~# cat file

hello

root@DESKTOP-5HM2HTK:~# cat file3

hello

root@DESKTOP-5HM2HTK:~# cmp file file3

root@DESKTOP-5HM2HTK:~#

С помощью ключа -c(или -b) мы можем узнать сами различающиеся знаки и их код asci в восьмиричном представлении:

root@DESKTOP-5HM2HTK:~# cmp -c file1 file2

file1 file2 differ: byte 6, line 2 is 127 W 121 Q

Ключ –l показывает все байты файлов, которые находятся после первого найденного различия, и первые различающиеся байты также выводятся. Байты представлены в восьмиричной системе:

root@DESKTOP-5HM2HTK:~# cmp -l file1 file2

6 127 121

7 157 162

8 162 157

9 154 142

10 144 12

11 12 106

12 114 162

13 145 157

14 163 147

15 163 12

cmp: EOF on file2 after byte 15

root@DESKTOP-5HM2HTK:~#

При ключе –s не будет напечатано ничего, даже если различие есть. С помощью него мы просто можем узнать, одинаковые файлы или нет:

root@DESKTOP-5HM2HTK:~# cat file

hello

root@DESKTOP-5HM2HTK:~# cat file3

hello

root@DESKTOP-5HM2HTK:~# cat file1

Hell

World

Less

root@DESKTOP-5HM2HTK:~# cmp -s file file1

root@DESKTOP-5HM2HTK:~# echo $?

1

root@DESKTOP-5HM2HTK:~# cmp -s file file3

root@DESKTOP-5HM2HTK:~# echo $?

0

1. **comm**

С помощью команды comm идёт также сравнение двух файлов. Важно, чтобы файлы были отсортированы. Вывод команды состоит из трёх столбцов: в первом пишутся те строчки, которые есть только в первом переданном файле; во втором – строчки, которые есть только во втором переданном файле; в третьем – строчки, которые присутствуют в обоих переданных файлах:

root@DESKTOP-5HM2HTK:~# sort file1 >file1\_sort

root@DESKTOP-5HM2HTK:~# sort file2 >file2\_sort

root@DESKTOP-5HM2HTK:~# cat file1

Hell

World

Less

root@DESKTOP-5HM2HTK:~# cat file1\_sort

Hell

Less

World

root@DESKTOP-5HM2HTK:~# cat file2

Hell

World

Frog

root@DESKTOP-5HM2HTK:~# cat file2\_sort

Frog

Hell

World

root@DESKTOP-5HM2HTK:~# comm file1\_sort file2\_sort

Frog

Hell

Less

World

root@DESKTOP-5HM2HTK:~#

С помощью ключей -1, -2 и -3 можно исключать первый, второй или третий столбец. Таким образом, можно, например, увидеть только те строчки, в которых файлы сходятся:

root@DESKTOP-5HM2HTK:~# comm -1 -2 file1\_sort file2\_sort

Hell

World

root@DESKTOP-5HM2HTK:~#

1. **diff.**

Разберём подробно команду diff. Это ещё одна утилита для сравнения ДВУХ файлов, и выводом её является список действий, который позволит сделать из первого переданного файла второй файл. Сравнение идёт построчно:

 a означает добавлено (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) add), d — удалено, с — изменено. Перед буквами a, d или c стоят номера строк исходного файла, после них — номера строк конечного файла

root@DESKTOP-5HM2HTK:~# cat file1

Hell

Ebcdic

World

Less

root@DESKTOP-5HM2HTK:~# cat file2

Hell

Ascii

World

root@DESKTOP-5HM2HTK:~# diff file1 file2

2c2

< Ebcdic

---

> Ascii

4d3

< Less

С ключом –u список станет более удобным и понятным:

root@DESKTOP-5HM2HTK:~# diff -u file1 file2

--- file1 2021-02-11 23:08:30.819887300 +0400

+++ file2 2021-02-11 23:09:20.387317100 +0400

@@ -1,4 +1,3 @@

Hell

-Ebcdic

+Ascii

World

-Less

Если файлы одинаковые, то выведено ничего не будет:

root@DESKTOP-5HM2HTK:~# cat file

hello

root@DESKTOP-5HM2HTK:~# cat file3

hello

root@DESKTOP-5HM2HTK:~# diff file file3

root@DESKTOP-5HM2HTK:~#

С помощью ключа –i игнорируется регистр букв:

root@DESKTOP-5HM2HTK:~# cat file

Hello

root@DESKTOP-5HM2HTK:~# cat file3

hello

root@DESKTOP-5HM2HTK:~# diff -i file file3

root@DESKTOP-5HM2HTK:~# diff file file3

1c1

< Hello

---

> hello

root@DESKTOP-5HM2HTK:~#

С помощью ключа -b игнорируются любые различия в пробелах (главное, чтобы были сами пробелы/табуляции):

root@DESKTOP-5HM2HTK:~# cat file1

Hell Grow

Ebcdic

WorldOPS

Less

root@DESKTOP-5HM2HTK:~# cat file2

Hell Grow

Ascii

World OPS

root@DESKTOP-5HM2HTK:~# diff -b file1 file2

2,4c2,3

< Ebcdic

< WorldOPS

< Less

---

> Ascii

> World OPS

root@DESKTOP-5HM2HTK:~# diff file1 file2

1,4c1,3

< Hell Grow

< Ebcdic

< WorldOPS

< Less

---

> Hell Grow

> Ascii

> World OPS

root@DESKTOP-5HM2HTK:~#

При использовании ключа -w будут проигнорированы все пробелы(табуляции):

root@DESKTOP-5HM2HTK:~# diff -w file1 file4

2c2

< Ebcdic

---

> Ascii

4d3

< Less

root@DESKTOP-5HM2HTK:~#

\*Ключи –с и –С NUM – число строк контекста\*

root@DESKTOP-5HM2HTK:~# diff -c file1 file2

\*\*\* file1 2021-02-13 15:36:37.071318600 +0400

--- file2 2021-02-13 15:34:29.478136600 +0400

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 1,4 \*\*\*\*

! Hell World

! Ebcdic

World

- Less

--- 1,3 ----

! Hell

! Ascii

World

root@DESKTOP-5HM2HTK:~# diff -C0 file1 file2

\*\*\* file1 2021-02-13 15:36:37.071318600 +0400

--- file2 2021-02-13 15:34:29.478136600 +0400

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 1,2 \*\*\*\*

! Hell World

! Ebcdic

--- 1,2 ----

! Hell

! Ascii

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 4 \*\*\*\*

- Less

--- 3 ----

\*Ключи –e\*

root@DESKTOP-5HM2HTK:~# cat file1

Hell World

Ebcdic

World

Less

root@DESKTOP-5HM2HTK:~# cat file2

Hell

Ascii

World

root@DESKTOP-5HM2HTK:~# diff -e file1 file2

5d

1,3c

Hell

Ascii

.

root@DESKTOP-5HM2HTK:~#

Также с помощью diff можно сравнивать директории:

root@DESKTOP-5HM2HTK:~# diff Desktop Top

Common subdirectories: Desktop/Frog and Top/Frog

Only in Desktop: Frog1

Common subdirectories: Desktop/Missing and Top/Missing

Only in Top: Missing1

root@DESKTOP-5HM2HTK:~#

-------------------------------------------------------------------------

(были созданы пустые файлы file и один файл file1)

root@DESKTOP-5HM2HTK:~# diff Desktop Top

Common subdirectories: Desktop/Frog and Top/Frog

Only in Desktop: Frog1

Common subdirectories: Desktop/Missing and Top/Missing

Only in Top: Missing1

Only in Top: file1

root@DESKTOP-5HM2HTK:~#

-------------------------------------------------------------------------

(один из file был изменён)

root@DESKTOP-5HM2HTK:~# diff Desktop Top

Common subdirectories: Desktop/Frog and Top/Frog

Only in Desktop: Frog1

Common subdirectories: Desktop/Missing and Top/Missing

Only in Top: Missing1

diff Desktop/file Top/file

0a1

> q

Only in Top: file1

Ключ –r позволяет проводить сравнение рекурсивно:

root@DESKTOP-5HM2HTK:~# ls Desktop/

Frog Frog1 Missing file file1

root@DESKTOP-5HM2HTK:~# ls Top/

Frog Missing Missing1 file file1

root@DESKTOP-5HM2HTK:~# ls Desktop/Frog

file1

root@DESKTOP-5HM2HTK:~# diff -r Desktop Top

Only in Desktop/Frog: file1

Only in Desktop: Frog1

Only in Top/Missing: file2

Only in Top: Missing1

diff -r Desktop/file Top/file

0a1

> q

diff -r Desktop/file1 Top/file1

1d0

< q

root@DESKTOP-5HM2HTK:~#

С помощью ключа –S можно начать сравнение каталогов с указанного объекта:

root@DESKTOP-5HM2HTK:~# diff -S Missing Top Desktop

Common subdirectories: Top/Missing and Desktop/Missing

Only in Top: Missing1

diff -S Missing Top/file Desktop/file

1d0

< q

diff -S Missing Top/file1 Desktop/file1

0a1

> q

root@DESKTOP-5HM2HTK:~#

------------------------------------------------------------------

root@DESKTOP-5HM2HTK:~# diff -r -S Missing Top Desktop

Only in Top/Missing: file2

Only in Top: Missing1

diff -r -S Missing Top/file Desktop/file

1d0

< q

diff -r -S Missing Top/file1 Desktop/file1

0a1

> q

root@DESKTOP-5HM2HTK:~#

Ключ –s по сути работает также как обычный diff, однако если файлы одинаковые, то он об этом сообщит:

root@DESKTOP-5HM2HTK:~# diff -s file1 file2

1,3c1,2

< Hell World

<

< Ebcdic

---

> Hell

> Ascii

5d3

< Less

root@DESKTOP-5HM2HTK:~#

root@DESKTOP-5HM2HTK:~# diff -s file file3

Files file and file3 are identical

Ключ –q лишь сообщает нам, различаются ли файлы или нет:

root@DESKTOP-5HM2HTK:~# diff -q file file3

root@DESKTOP-5HM2HTK:~# diff -q file1 file2

Files file1 and file2 differ

root@DESKTOP-5HM2HTK:~#

\*компиляция файла с директивами препроцессора языка Си с помощью ключа –D строчка\*

1. **patch**

С помощью данной команды происходит преобразование одного файла в другой при помощи инструкций, полученных при помощи команды diff –u:

root@DESKTOP-5HM2HTK:~# cat file1

Hell World

Ebcdic

World

Pullup

root@DESKTOP-5HM2HTK:~# cat file2

Hell

Ascii

World

root@DESKTOP-5HM2HTK:~# diff -u file1 file2 | patch

patching file file1

root@DESKTOP-5HM2HTK:~# diff -s file1 file2

Files file1 and file2 are identical

root@DESKTOP-5HM2HTK:~#

При использовании ключа –b создастся резервная копия, которая будет являться оригиналом первого файла:

root@DESKTOP-5HM2HTK:~# diff -u file1 file2 | patch -b

patching file file1

root@DESKTOP-5HM2HTK:~# ls

Desctop file file1 file1.orig file2 file3 file4 upgrade

root@DESKTOP-5HM2HTK:~# cat file1.orig

Hell World

Ebcdic

World

Pullup

root@DESKTOP-5HM2HTK:~#

С patch можно использовать любой ключ (или почти любой), который был использован при составлении инструкции с diff:

root@DESKTOP-5HM2HTK:~# diff -c file1 file2 | patch -c

patching file file1

root@DESKTOP-5HM2HTK:~# diff -s file1 file2

Files file1 and file2 are identical

root@DESKTOP-5HM2HTK:~#

1. **grep**

Команда grep осуществляет поиск нужных элементов в файле (файлах). Рассмотрим несколько способов её применения.

Без ключей он выполняет самую банальную рассмотренную нами в первой (№2) лабораторной работе функцию – ищет заданный нами текст в файле, и выдаёт все строчки, где был найден данный текст, подчёркивая его самого красным. Однако, это если после команды мы напишем обычный текст. Если мы перед ним поставим знак ^ и заключим всё это в кавычки, то теперь grep будет искать строчки, начинающиеся конкретно с этого текста. Если после нашего текста мы поставим знак $ и также заключим всё это в скобки, то grep будет искать строчки, которые заканчиваются данным текстом:

root@DESKTOP-5HM2HTK:~# cat file1

Hell World

Ebcdic

World

World Hell

Pullup

root@DESKTOP-5HM2HTK:~# grep 'World' <file1

Hell World

World

World Hell

root@DESKTOP-5HM2HTK:~# grep '^World' <file1

World

World Hell

root@DESKTOP-5HM2HTK:~# grep 'World$' <file1

Hell World

World

root@DESKTOP-5HM2HTK:~# grep '^World$' <file1

World

root@DESKTOP-5HM2HTK:~#

Также можно осуществлять поиск по крайним элементам, между которым поставить точку или точку и звёздочку. Точка означает ровно один знак, а точка и звёздочка – любою (даже пустую) последовательность:

root@DESKTOP-5HM2HTK:~# grep 'W.l' <file1

root@DESKTOP-5HM2HTK:~# grep 'W..l' <file1

Hell World

World

World Hell

root@DESKTOP-5HM2HTK:~# grep 'W..\*l' <file1

Hell World

World

World Hell

root@DESKTOP-5HM2HTK:~#

Ключ –с выдаёт, в скольких строках встречается указанный текст:

root@DESKTOP-5HM2HTK:~# grep -c 'W...\*l' <file1

3

root@DESKTOP-5HM2HTK:~#

Ключ –l просто указывает, в каких файлах содержится указанный текст:

root@DESKTOP-5HM2HTK:~# grep -l 'World' file1 file2 file

file1

file2

root@DESKTOP-5HM2HTK:~#

С помощью ключа –v наоборот можно увидеть все строчки, в которых данный текст отсутствует:

root@DESKTOP-5HM2HTK:~# grep -v 'World' file1 file2 file

file1:Ebcdic

file1:

file1:Pullup

file2:Hell

file2:Ascii

file:hello

root@DESKTOP-5HM2HTK:~#

1. **wc**

Команда wc также была поверхностно разобрана в первой лабораторной работе (№2). Без ключей она показывает количество строк, слов и байтов в файле/файлах/тексте:

root@DESKTOP-5HM2HTK:~# wc file1 file

6 7 46 file1

1 1 6 file

7 8 52 total

root@DESKTOP-5HM2HTK:~#

Ключ –с показывает количество байт, из которого состоит текст файла, ключ –m – количество знаков в тексте. Если все знаки являются знаками ascii, то это количество одинаково:

root@DESKTOP-5HM2HTK:~# cat file1

Нell World

Ebcdic

World

World Hell

Pullup

root@DESKTOP-5HM2HTK:~# cat file

hello

root@DESKTOP-5HM2HTK:~# wc -c file1 file

46 file1

6 file

52 total

root@DESKTOP-5HM2HTK:~# wc -m file1 file

46 file1

6 file

52 total

root@DESKTOP-5HM2HTK:~#

А в этом примере одна буква была заменена на русскую:

root@DESKTOP-5HM2HTK:~# wc -m file1

46 file1

root@DESKTOP-5HM2HTK:~# wc -c file1

47 file1

С помощью ключа –l можно узнать количество строк в файле, -w – количество слов:

root@DESKTOP-5HM2HTK:~# wc -l file1 file

6 file1

1 file

7 total

root@DESKTOP-5HM2HTK:~# wc -w file1 file

7 file1

1 file

8 total

root@DESKTOP-5HM2HTK:~#

Ключ –L печатает длину самой большой строки в файле и в конце пишет самое большое среди всех значений:

root@DESKTOP-5HM2HTK:~# wc -L file1 file file2

11 file1

5 file

7 file2

11 total

root@DESKTOP-5HM2HTK:~#

1. **dd**

С помощью команды dd осуществляется копирование данных из одного файла в другой:

if='файл, с которого копируются данные'

of=’файл, куда копируются данные’

root@DESKTOP-5HM2HTK:~# dd if=file1 of=file5

0+1 records in

0+1 records out

45 bytes copied, 0.0011064 s, 40.7 kB/s

root@DESKTOP-5HM2HTK:~# diff -s file1 file5

Files file1 and file5 are identical

root@DESKTOP-5HM2HTK:~#

Команде можно задавать некоторые характеристики. Например, с помощью bs можно задать количество байт в блоке при передаче данных, с помощью count можно указать само количество блоков данных, которое мы хоти передать (по умолчанию – один):

root@DESKTOP-5HM2HTK:~# dd if=file1 of=file5 bs=5 count=3

3+0 records in

3+0 records out

15 bytes copied, 0.0011414 s, 13.1 kB/s

root@DESKTOP-5HM2HTK:~# cat file5

Hell World

Ebcdroot@DESKTOP-5HM2HTK:~#

С помощью skip можно задать количество блоков, которое будет пропущено перед началом передачи файлов:

root@DESKTOP-5HM2HTK:~# dd if=file1 of=file5 skip=1

dd: file1: cannot skip to specified offset

0+0 records in

0+0 records out

0 bytes copied, 0.0011038 s, 0.0 kB/s

root@DESKTOP-5HM2HTK:~# dd if=file1 of=file5 bs=5 count=3 skip=1

3+0 records in

3+0 records out

15 bytes copied, 0.0017044 s, 8.8 kB/s

root@DESKTOP-5HM2HTK:~# cat file5

World

Ebcdic

root@DESKTOP-5HM2HTK:~#

Благодаря conv можно указывать различные параметры передачи файлов. Например, изменить строчные буквы на заглавные с помощью ucase, или заглавные на строчные с помощью lcase:

root@DESKTOP-5HM2HTK:~# dd if=file1 of=file5 conv=ucase

0+1 records in

0+1 records out

45 bytes copied, 0.0016616 s, 27.1 kB/s

root@DESKTOP-5HM2HTK:~# cat file5

HELL WORLD

EBCDIC

WORLD

WORLD HELL

PULLUP

root@DESKTOP-5HM2HTK:~# dd if=file1 of=file5 conv=lcase

0+1 records in

0+1 records out

45 bytes copied, 0.0009926 s, 45.3 kB/s

root@DESKTOP-5HM2HTK:~# cat file5

hell world

ebcdic

world

world hell

pullup

root@DESKTOP-5HM2HTK:~#

Также можно производить преобразования из одной кодировки в другую.

1. **tee**

Команда tee очень похожа на команду cat, однако при использовании tee данные со стандартного ввода копируются не только в файлы-аргументы, но ещё и на стандартный вывод:

root@DESKTOP-5HM2HTK:~# touch file4

root@DESKTOP-5HM2HTK:~# grep 'Hell' file1 | cat >file4

root@DESKTOP-5HM2HTK:~# rm file4

root@DESKTOP-5HM2HTK:~# touch file4

root@DESKTOP-5HM2HTK:~# grep 'Hell' file1 | tee >file4

Hell World

World Hell

root@DESKTOP-5HM2HTK:~#

Данные со стандартного вывод можно задействовать в конвейере. Таким образом мы можем сделать два действия одновременно – записать данные файл и пустить их на поток конвейера:

root@DESKTOP-5HM2HTK:~# touch file4

root@DESKTOP-5HM2HTK:~# grep 'Hell' file1 | tee file4 | grep 'd.H'

World Hell

root@DESKTOP-5HM2HTK:~#

С помощью ключа –a данные будут записаны ниже тех, что уже есть в файле (по умолчанию данные перезаписываются).

1. **tr**

С помощью команды tr можно производить замену указанных знаков на другие, также указанные:

root@DESKTOP-5HM2HTK:~# cat file1

Hell World

Ebcdic

World

World Hell

Pullup

root@DESKTOP-5HM2HTK:~# tr abc bcd <file1

Hell World

Ecddid

World

World Hell

Pullup

root@DESKTOP-5HM2HTK:~#

При использовании ключа –с все знаки текста из первого набора не меняются, а другие заменяются на знак из второго текста:

hello-root@DESKTOP-5HM2HTK:~# cat file1

Hell World

Ebcdic

World

World Hell

Pullup

root@DESKTOP-5HM2HTK:~# tr -c abcd - <file1

---------d--bcd-c--------d-----d-------------root@DESKTOP-5HM2HTK:~# tr -c abcd GP <file1

PPPPPPPPPdPPbcdPcPPPPPPPPdPPPPPdPPPPPPPPPPPPProot@DESKTOP-5HM2HTK:~#

С помощью ключа –d можно удалить указанные знаки в тексте:

root@DESKTOP-5HM2HTK:~# tr -d abcd <file1

Hell Worl

Ei

Worl

Worl Hell

Pullup

root@DESKTOP-5HM2HTK:~#

С помощью ключа –s удаляются все подряд идущие знаки из указанного набора и остаётся только один из них:

root@DESKTOP-5HM2HTK:~# tr -s bcd

bbccdd

bcd

root@DESKTOP-5HM2HTK:~#

1. **od/xxd**

С помощью команд od и xxd можно получить восьмиричное и шестнадцатиричное представление файлов соответственно:

root@DESKTOP-5HM2HTK:~# cat file1

Hell World

Ebcdic

World

World Hell

Pullup

root@DESKTOP-5HM2HTK:~# od file1

0000000 062510 066154 053440 071157 062154 042412 061542 064544

0000020 005143 020040 053412 071157 062154 053412 071157 062154

0000040 044040 066145 005154 072520 066154 070165 000012

0000055

(восьмиричные двухбайтовые единицы)

root@DESKTOP-5HM2HTK:~# xxd file1

00000000: 4865 6c6c 2057 6f72 6c64 0a45 6263 6469 Hell World.Ebcdi

00000010: 630a 2020 0a57 6f72 6c64 0a57 6f72 6c64 c. .World.World

00000020: 2048 656c 6c0a 5075 6c6c 7570 0a Hell.Pullup.

root@DESKTOP-5HM2HTK:~#

С помощью ключей –j и –N можно начать с указанного байта и напечатать также указанное количество соответственно:

root@DESKTOP-5HM2HTK:~# od -j 10 file1

0000012 042412 061542 064544 005143 020040 053412 071157 062154

0000032 053412 071157 062154 044040 066145 005154 072520 066154

0000052 070165 000012

0000055

root@DESKTOP-5HM2HTK:~# od -N 5 file1

0000000 062510 066154 000040

0000005

root@DESKTOP-5HM2HTK:~#

Также можно переводить данное представление в, например, в десятиричные двухбайтовые единицы или в восьмиричные байты:

root@DESKTOP-5HM2HTK:~# od -b file1

0000000 110 145 154 154 040 127 157 162 154 144 012 105 142 143 144 151

0000020 143 012 040 040 012 127 157 162 154 144 012 127 157 162 154 144

0000040 040 110 145 154 154 012 120 165 154 154 165 160 012

0000055

root@DESKTOP-5HM2HTK:~# od -s file1

0000000 25928 27756 22304 29295 25708 17674 25442 26980

0000020 2659 8224 22282 29295 25708 22282 29295 25708

0000040 18464 27749 2668 30032 27756 28789 10

0000055

root@DESKTOP-5HM2HTK:~#

1. **sed**

Команда sed является редактором потока. Рассмотрим немного функционал данной команды.

С помощью конструкции sed ‘/”текст”/d’ можно удалить строчки, содержащие указанный текст, из файла:

root@DESKTOP-5HM2HTK:~# cat file1

Hell World

Ebcdic

World

World Hell

Pullup

root@DESKTOP-5HM2HTK:~# sed '/World/d' file1

Ebcdic

Pullup

root@DESKTOP-5HM2HTK:~#

С помощью неё очень удобно удалять пустые строчки:

root@DESKTOP-5HM2HTK:~# sed '/^\s\*$/d' file1

Hell World

Ebcdic

World

World Hell

Pullup

root@DESKTOP-5HM2HTK:~#

^ - начало строки; \s – любой пробельный символ; \* - любое количество повторений прошлого знака; $ - конец строки.

С помощью конструкции ‘s/”текст1”/”текст2”/g’ происходит замена первой комбинации знаков на вторую:

s — замена; g — глобально

root@DESKTOP-5HM2HTK:~# sed 's/World/s/g' file1

Hell s

Ebcdic

s

s Hell

Pullup

root@DESKTOP-5HM2HTK:~#

(заместо слеша могут быть любые знаки) :

root@DESKTOP-5HM2HTK:~# sed 'sQWorldQsQg' file1

Hell s

Ebcdic

s

s Hell

Pullup

root@DESKTOP-5HM2HTK:~#

При использовании конструкции sed ‘a\”текст”’ указанный текст будет вставлен перед всеми строчками, вхождение которых начинается с обратного слеша. При использовании конструкции sed ‘с\”текст”’ строчки, вхождение которых начинается с обратного слеша, будут заменены указанным текстом, а остальные строчки будут стёрты:

root@DESKTOP-5HM2HTK:~# sed 'a\rer' file1

Hell World

rer

Ebcdic

rer

rer

World

rer

World Hell

rer

Pullup

rer

root@DESKTOP-5HM2HTK:~# sed 'c\rer' file1

rer

rer

rer

rer

rer

rer

root@DESKTOP-5HM2HTK:~#

\*ключ –e добавляет команды в сценарий к набору выполняемых команд при обработке ввода\*

1. **awk**

Благодаря утилите awk мы можем извлекать из файла данные с заданными нами характеристиками. Например, с помощью него мы можем вывести все вторые и третьи слова строчек, в которых встречается слово “Frog”:

root@DESKTOP-5HM2HTK:~# cat file1

Hell World Frog

Ebcdic Hell World

World Frog Hell

World Hell Frog

Pullup World Hell

root@DESKTOP-5HM2HTK:~# awk '/'Frog'/{print $2,$3}' file1

World Frog

Frog Hell

Hell Frog

root@DESKTOP-5HM2HTK:~#

Так как awk – полноценный язык, то ему можно указывать программы, по которым он будет производить отбор данных:

root@DESKTOP-5HM2HTK:~# awk '/'^W'/{i=2; while (i<=3) {print $i i++}}' file1

Frog2

Hell3

Hell2

Frog3

root@DESKTOP-5HM2HTK:~#

Также можно выписывать слова чётных/нечётных строк (делящихся на 3,4,5,…):

root@DESKTOP-5HM2HTK:~# awk 'NR%2==0 {print $1}' file1

Ebcdic

World

Pullup

root@DESKTOP-5HM2HTK:~#

NR означает Number of Rows

Также можно регулировать длину напечатанных строк:

root@DESKTOP-5HM2HTK:~# awk 'length($0) >15' file1

Ebcdic Hell World

Pullup World Hell

root@DESKTOP-5HM2HTK:~#

1. **du**

Команда du предназначена для выдачи отчётов об использовании дискового пространства заданными файлами или директориями. По умолчанию без аргументов команда выдаёт отчёт о дисковом пространстве для текущего каталога:

root@DESKTOP-5HM2HTK:~/Desctop# du

16 .

root@DESKTOP-5HM2HTK:~# awk '/'^W'/{i=2; while (i<=1000) {print i i++}}' file1 >file4

root@DESKTOP-5HM2HTK:~# du file4

16 file4

root@DESKTOP-5HM2HTK:~# du

0 ./.local/share/nano

0 ./.local/share

0 ./.local

4 ./.ssh

0 ./Desctop

80 .

root@DESKTOP-5HM2HTK:~#

Цифры в начале строк – занимаемый размер в блоках по 1024 байт. Об этом нам может сказать ключ –h:

root@DESKTOP-5HM2HTK:~# du -h

0 ./.local/share/nano

0 ./.local/share

0 ./.local

4.0K ./.ssh

0 ./Desctop

80K .

root@DESKTOP-5HM2HTK:~#

Так как обычные текстовые файлы далеко не всегда занимают 1024 байта, мы можем воспользоваться ключом –b, которые выдают значение в байтах, а не в килобайтах:

root@DESKTOP-5HM2HTK:~# cat file

Hello

root@DESKTOP-5HM2HTK:~# du -b file

6 file

root@DESKTOP-5HM2HTK:~# du file

0 file

root@DESKTOP-5HM2HTK:~#

Но если файлы всё таки очень большие, то мы можем воспользоваться ключом –m, который считает размер в мегабайтах:

root@DESKTOP-5HM2HTK:~# awk '/'^W'/{i=2; while (i<=10000) {print i i++}}' file1 >file5

root@DESKTOP-5HM2HTK:~# du file5

2112 file5

root@DESKTOP-5HM2HTK:~# du -m file5

3 file5

root@DESKTOP-5HM2HTK:~#

Ключ –B’NUM’ указывает размер компонентов в желаемых единицах. Например \_-BM заменит –m.

Как мы могли заметить, без ключей du выдаёт размеры лишь каталогов, которые лежат ниже текущего каталога. С помощью ключа –a команда нам выдаст информацию ещё и о всех файлах, которые сможет найти ниже текущего каталога (и в самом текущем тоже):

root@DESKTOP-5HM2HTK:~# du -a

36 ./.bash\_history

4 ./.bashrc

0 ./.local/share/nano/search\_history

0 ./.local/share/nano

0 ./.local/share

0 ./.local

0 ./.motd\_shown

0 ./.profile

4 ./.ssh/known\_hosts

4 ./.ssh

12 ./.swp

4 ./.test.s.swp

4 ./.viminfo

0 ./Desctop/Frog

0 ./Desctop/file1

16 ./Desctop/file4

16 ./Desctop

0 ./file

0 ./file1

0 ./file2

0 ./file3

16 ./file4

96 .

root@DESKTOP-5HM2HTK:~#

Если вдруг вложенных директорий у нас просто куча и нам не требуется настолько глубоко лезть, то с помощью ключа --max-depth=’NUM’ можно ограничить уровень, до которого буде ‘копать’ du:

root@DESKTOP-5HM2HTK:~# mkdir Desctop/Frog/Root1

root@DESKTOP-5HM2HTK:~# nano Desctop/Frog/Root1/File

root@DESKTOP-5HM2HTK:~# du --max-depth=2

0 ./.local/share

0 ./.local

4 ./.ssh

0 ./Desctop/Frog

16 ./Desctop

2208 .

root@DESKTOP-5HM2HTK:~#

А если нам нужно избежать какого то конкретного компонента, то мы можем воспользоваться ключом –X:

root@DESKTOP-5HM2HTK:~# du -a -X -

file

Frog

36 ./.bash\_history

4 ./.bashrc

0 ./.local/share/nano/search\_history

0 ./.local/share/nano

0 ./.local/share

0 ./.local

0 ./.motd\_shown

0 ./.profile

4 ./.ssh/known\_hosts

4 ./.ssh

12 ./.swp

4 ./.test.s.swp

4 ./.viminfo

0 ./Desctop/file1

16 ./Desctop/file4

16 ./Desctop

0 ./file1

0 ./file2

0 ./file3

16 ./file4

2112 ./file5

2208 .

root@DESKTOP-5HM2HTK:~#

Вообще, он читает данные из файла, но при использовании – можно дать данные и со стандартного ввода.

Ключ –s вовсе обрубит нам всё лишнее и покажет лишь общую сумму текущей директории либо переданного компонента:

root@DESKTOP-5HM2HTK:~# du -s

2208 .

root@DESKTOP-5HM2HTK:~# du -s Desctop

16 Desctop

root@DESKTOP-5HM2HTK:~#

1. **df**

Команда df выдаёт отчёт о доступном и использованном месте на дисках:

root@DESKTOP-5HM2HTK:~# df

Filesystem 1K-blocks Used Available Use% Mounted on

rootfs 523740872 72058948 451681924 14% /

none 523740872 72058948 451681924 14% /dev

none 523740872 72058948 451681924 14% /run

none 523740872 72058948 451681924 14% /run/lock

none 523740872 72058948 451681924 14% /run/shm

none 523740872 72058948 451681924 14% /run/user

tmpfs 523740872 72058948 451681924 14% /sys/fs/cgroup

C:\ 523740872 72058948 451681924 14% /mnt/c

E:\ 452440604 185512780 266927824 42% /mnt/e

root@DESKTOP-5HM2HTK:~# df Desctop file1

Filesystem 1K-blocks Used Available Use% Mounted on

rootfs 523740872 72067288 451673584 14% /

rootfs 523740872 72067288 451673584 14% /

root@DESKTOP-5HM2HTK:~#

df с аргументом даёт отчёт по файловой системе, в которой находится данный элемент

Большинство ключей для du подходит и для df (например: -a,-k,-m):

root@DESKTOP-5HM2HTK:~# df -BM Desctop file1

Filesystem 1M-blocks Used Available Use% Mounted on

rootfs 511466M 70378M 441089M 14% /

rootfs 511466M 70378M 441089M 14% /

root@DESKTOP-5HM2HTK:~# df -a

Filesystem 1K-blocks Used Available Use% Mounted on

rootfs 523740872 72062880 451677992 14% /

none 523740872 72062880 451677992 14% /dev

sysfs 0 0 0 - /sys

proc 0 0 0 - /proc

devpts 0 0 0 - /dev/pts

none 523740872 72062880 451677992 14% /run

none 523740872 72062880 451677992 14% /run/lock

none 523740872 72062880 451677992 14% /run/shm

none 523740872 72062880 451677992 14% /run/user

binfmt\_misc 0 0 0 - /proc/sys/fs/binfmt\_misc

tmpfs 523740872 72062880 451677992 14% /sys/fs/cgroup

cgroup 0 0 0 - /sys/fs/cgroup/devices

C:\ 523740872 72062880 451677992 14% /mnt/c

E:\ 452440604 185512780 266927824 42% /mnt/e

root@DESKTOP-5HM2HTK:~#

\*Ключ –l выводит только данные о локальных файловых системах\*

При использовании ключа –T выдаётся тип каждой файловой системы. Ключом –x можно выбрать, данные каких файловых систем НЕ выводить:

root@DESKTOP-5HM2HTK:~# df -T

Filesystem Type 1K-blocks Used Available Use% Mounted on

rootfs lxfs 523740872 72064404 451676468 14% /

none tmpfs 523740872 72064404 451676468 14% /dev

none tmpfs 523740872 72064404 451676468 14% /run

none tmpfs 523740872 72064404 451676468 14% /run/lock

none tmpfs 523740872 72064404 451676468 14% /run/shm

none tmpfs 523740872 72064404 451676468 14% /run/user

tmpfs tmpfs 523740872 72064404 451676468 14% /sys/fs/cgroup

C:\ drvfs 523740872 72064404 451676468 14% /mnt/c

E:\ drvfs 452440604 185512780 266927824 42% /mnt/e

root@DESKTOP-5HM2HTK:~# df -T -x tmpfs

Filesystem Type 1K-blocks Used Available Use% Mounted on

rootfs lxfs 523740872 72063428 451677444 14% /

C:\ drvfs 523740872 72063428 451677444 14% /mnt/c

E:\ drvfs 452440604 185512780 266927824 42% /mnt/e

root@DESKTOP-5HM2HTK:~#

1. **file**

С помощью команды file мы можем узнать формат объекта:

root@DESKTOP-5HM2HTK:~# file file1 Desctop prog.c

file1: ASCII text

Desctop: directory

prog.c: C source, ASCII text

root@DESKTOP-5HM2HTK:~#

При ключе –f переданный файл будет использован как файл, содержащий аргументы для команды. Аргументы читаются построчно:

root@DESKTOP-5HM2HTK:~# cat names

file

file1

file2 file3

file4

Desctop

русский

root@DESKTOP-5HM2HTK:~# file -f names

file: ASCII text

file1: ASCII text

file2 file3: cannot open `file2 file3' (No such file or directory)

file4: ASCII text

Desctop: directory

русский: UTF-8 Unicode text

root@DESKTOP-5HM2HTK:~#

С помощью ключа –i будет показана некоторая дополнительная информация:

root@DESKTOP-5HM2HTK:~# file -i names Desctop prog.c русский

names: text/plain; charset=us-ascii

Desctop: inode/directory; charset=binary

prog.c: text/x-c; charset=us-ascii

русский: text/plain; charset=utf-8

root@DESKTOP-5HM2HTK:~#

1. **split**

Благодаря команде split можно разбить один файл на несколько. Сначала пишется ключ –n (если мы хотим разбить на определённое количество файлов), затем количество файлов, на который будет разбит наш переданный файл, затем сам файл:

root@DESKTOP-5HM2HTK:~# split -n 3 file1

root@DESKTOP-5HM2HTK:~# wc x\*

1 6 29 xaa

3 5 29 xab

2 5 29 xac

6 16 87 total

root@DESKTOP-5HM2HTK:~# cat file1

Hell World Frog

Ebcdic Hell World

World Frog Hell

World Hell Frog

Pullup World Hell

root@DESKTOP-5HM2HTK:~# cat xaa

Hell World Frog

Ebcdic Hell Wroot@DESKTOP-5HM2HTK:~# cat xab

orld

World Frog Hell

Worldroot@DESKTOP-5HM2HTK:~# cat xac

Hell Frog

Pullup World Hell

root@DESKTOP-5HM2HTK:~#

По умолчанию префиксом у производных файлов является x. Префикс можно указать самим через пробел:

root@DESKTOP-5HM2HTK:~# split -n 3 file1 proizvodnaya

root@DESKTOP-5HM2HTK:~# ls

Desctop file file1 file2 file3 proizvodnayaaa proizvodnayaab proizvodnayaac

root@DESKTOP-5HM2HTK:~#

Если вдруг производных очень много, то с помощью ключа –a можно указывать количество суффиксов в их названии:

root@DESKTOP-5HM2HTK:~# split -a 4 -n 3 file1 qwert

root@DESKTOP-5HM2HTK:~# ls

Desctop file file1 file2 file3 qwertaaaa qwertaaab qwertaaac

root@DESKTOP-5HM2HTK:~#

С помощью ключа –b можно разбивать файл не по количеству производных файлов, а по количеству байтов в файлах:

root@DESKTOP-5HM2HTK:~# split -b 30 file1

root@DESKTOP-5HM2HTK:~# wc x\*

1 6 30 xaa

3 6 30 xab

2 5 27 xac

6 17 87 total

root@DESKTOP-5HM2HTK:~#

Ключ –l позволяет контролировать число строк в производных файлах:

root@DESKTOP-5HM2HTK:~# split -l 2 file1

root@DESKTOP-5HM2HTK:~# cat xaa

Hell World Frog

Ebcdic Hell World

root@DESKTOP-5HM2HTK:~# cat xab

World Hell Frog

root@DESKTOP-5HM2HTK:~# cat xac

Pullup World Hell

root@DESKTOP-5HM2HTK:~# cat file1

Hell World Frog

Ebcdic Hell World

World Hell Frog

Pullup World Hell

root@DESKTOP-5HM2HTK:~#

1. **mktemp**

Утилита mktemp позволяет нам создать пустой временный файл с произвольным именем (по умолчанию в папке tmp):

root@DESKTOP-5HM2HTK:~# mktemp

/tmp/tmp.MCqbByNmaE

root@DESKTOP-5HM2HTK:~# file /tmp/tmp.MCqbByNmaE

/tmp/tmp.MCqbByNmaE: empty

root@DESKTOP-5HM2HTK:~#

Можно создать файл с частично произвольным именем, поставив в конце знаки X (не меньше трёх!), которые будут заменены на случайные:

root@DESKTOP-5HM2HTK:~# mktemp ArtyomXXX.c

ArtyomzvV.c

root@DESKTOP-5HM2HTK:~#

С ключом –u будет просто выдано случайное название, файл создан не будет:

root@DESKTOP-5HM2HTK:~# mktemp -u

/tmp/tmp.omE6ycWYsN

root@DESKTOP-5HM2HTK:~# rm /tmp/tmp.omE6ycWYsN

rm: cannot remove '/tmp/tmp.omE6ycWYsN': No such file or directory

root@DESKTOP-5HM2HTK:~#

С помощью ключа –d можно также создавать временные директории:

root@DESKTOP-5HM2HTK:~# mktemp -d

/tmp/tmp.LHJpNHJ5uM

root@DESKTOP-5HM2HTK:~# file /tmp/tmp.LHJpNHJ5uM

/tmp/tmp.LHJpNHJ5uM: directory

root@DESKTOP-5HM2HTK:~# mktemp -d ArtyomXXX

ArtyomxeF

root@DESKTOP-5HM2HTK:~# file ArtyomxeF

ArtyomxeF: directory

root@DESKTOP-5HM2HTK:~#

Команда mktemp в основном используется для создания временных файлов / каталогов со случайным именем.

Таким образом, нам не нужно выяснять имена.

Так как mktemp рандомизирует имена, столкновения имен не будет.

Кроме того, mktemp создает файл / каталог безопасно с разрешением 600, поэтому другие пользователи не могут получить к нему доступ.

1. **find**

Благодаря команде find мы можем осуществлять поиск компонентов по имени и другим свойствам. По умолчанию find выдаёт всё, что видит ниже текущей директории:

root@DESKTOP-5HM2HTK:~# find

.

./.bash\_history

./.bashrc

./.local

./.local/share

./.local/share/nano

./.local/share/nano/search\_history

./.motd\_shown

./.profile

./.ssh

./.ssh/known\_hosts

./.swp

./.test.s.swp

./.viminfo

./Desctop

./Desctop/Frog

./Desctop/Frog/file1

./file

./file1

./file2

./file3

root@DESKTOP-5HM2HTK:~#

С ключом –ls, команда find выдаёт более полный список информации о файлах (инодный номер, размер в килобайтах, режим защиты, количество жестких ссылок, пользователь, группа, размер в байтах и время модификации).

Ключ –type задаёт тип искомых файлов ( f=файл, d=каталог (directory), l=ссылка (link), p=канал (pipe), s=сокет), а ключ –iname ищет файлы, имена которых соответствуют данному шаблону (есть ещё –name, но он регистрочувствителен):

root@DESKTOP-5HM2HTK:~# find /usr/ -type f -iname 'help.c\*'

/usr/lib/python3/dist-packages/jwt/\_\_pycache\_\_/help.cpython-38.pyc

/usr/lib/python3/dist-packages/requests/\_\_pycache\_\_/help.cpython-38.pyc

/usr/share/gnupg/help.ca.txt

/usr/share/gnupg/help.cs.txt

root@DESKTOP-5HM2HTK:~# find /usr/ -type d -iname 'big\*'

/usr/lib/x86\_64-linux-gnu/perl/5.30.0/Math/BigInt

/usr/lib/x86\_64-linux-gnu/perl/5.30.0/auto/Math/BigInt

/usr/share/perl/5.30.0/Math/BigFloat

/usr/share/perl/5.30.0/Math/BigInt

root@DESKTOP-5HM2HTK:~#

С помощью ключа –delete найденные компоненты удаляются:

root@DESKTOP-5HM2HTK:~# mktemp

/tmp/tmp.ey90XRBeVS

root@DESKTOP-5HM2HTK:~# find /tmp/ -delete -iname 'tmp.ey\*VS'

root@DESKTOP-5HM2HTK:~# find /tmp/tmp.ey90XRBeVS

find: ‘/tmp/tmp.ey90XRBeVS’: No such file or directory

root@DESKTOP-5HM2HTK:~#

Каталог tmp он тоже удалил.

Непустые директории не удаляет.

Ключ –size осуществляет поиск по размеру компонента. По умолчанию размер в блоках по 512 мегабайт, если после числа написать букву 'c’, размер будет переведён в байты:

root@DESKTOP-5HM2HTK:~# wc file1

5 12 71 file1

root@DESKTOP-5HM2HTK:~# find -type f -size 71c

./file1

root@DESKTOP-5HM2HTK:~#

1. **sort**

С помощью команды sort мы можем упорядочить текст или передать файл, в котором строчки будут переупорядочены лексикографически по первому элементу строки:

root@DESKTOP-5HM2HTK:~# cat file1

Hell World Frog

Ebcdic Hell World

World Hell Frog

Pullup World Hell

root@DESKTOP-5HM2HTK:~# sort file1

Ebcdic Hell World

Hell World Frog

Pullup World Hell

World Hell Frog

root@DESKTOP-5HM2HTK:~#

Ключ –r делает ревёрс результата, ключ –b игнорирует пробелы:

root@DESKTOP-5HM2HTK:~# sort -b -r file1

Pullup World Hell

Hell World Frog

Fight Like a Brave

Ebcdic Hell World

1World Hell Frog

root@DESKTOP-5HM2HTK:~#

Ключ –n предназначен для сравнения чисел с помощью ключа –k можно выбирать конкретные поля для сравнения:

root@DESKTOP-5HM2HTK:~# cat numbers

123

26

985

100 -25 1000

1024

12

-54

265

100 10000 200

root@DESKTOP-5HM2HTK:~# sort numbers

-54

0

100 -25 1000

100 10000 20

1024

12

123

26

265

985

root@DESKTOP-5HM2HTK:~# sort -n numbers

-54

0

12

26

100 -25 1000

100 10000 20

123

265

985

1024

root@DESKTOP-5HM2HTK:~# sort -n -k2,2 numbers

100 -25 1000

-54 0

265 13

123 36

26 42

0 224 345 16

1024 224 935

12 1000

985 1000

100 10000 20

root@DESKTOP-5HM2HTK:~# sort -n -k2,4 numbers

100 -25 1000

-54 0

265 13

123 36

26 42

0 224 345 16

1024 224 935

12 1000 -5

985 1000

100 10000 20

root@DESKTOP-5HM2HTK:~#

1. **cut**

Команда cut позволяет выделить столбцы текста из файла. Выделение происходит по байтам(-b), по знакам(-c) или слова(-f). При делении на слова необходимо указать разделитель (по умолчанию – табуляция).

С русскими буквами ломается. По сути должен выдавать при ключе –с саму букву, а при –b – ничего не выводить (как мне кажется). А на практике в обоих случаях ничего не вывел.

root@DESKTOP-5HM2HTK:~# cat file1

Hell World Frog

Fight Like A Brave

Ebcdic Hell World

1World Hell Frog

Pullup World Hell

root@DESKTOP-5HM2HTK:~# cut -b8 file1

r

i

H

H

W

root@DESKTOP-5HM2HTK:~# cut -c8 file1

r

i

H

H

W

root@DESKTOP-5HM2HTK:~# cut -f 2,3 -d" " file1

World Frog

Like A

Hell World

Hell Frog

World Hell

root@DESKTOP-5HM2HTK:~#

При ключе –s не печатаются строки, не содержащие разделителей:

root@DESKTOP-5HM2HTK:~# cut -f 1 -d" " -s file1

Hell

Fight

Ebcdic

1World

Pullup

root@DESKTOP-5HM2HTK:~# cut -f 1 -d" " file1

Hell

aaaaaaaaaaaaaaaaa

Fight

Ebcdic

1World

Pullup

root@DESKTOP-5HM2HTK:~#

Это может быть также пустая строка.

1. **paste**

Благодаря команде paste мы можем склеивать файлы построчно (к строке первого файла приписать строку второго файла):

root@DESKTOP-5HM2HTK:~# cat file1

Hell World Frog

aaaaaaaaaaaaaaaaa

Fight Like A Brave

Ebcdic Hell World

1World Hell Frog

Pullup World Hell

root@DESKTOP-5HM2HTK:~# cat part2

World Frog

Like A

Hell World

Hell Frog

root@DESKTOP-5HM2HTK:~# paste file1 part2

Hell World Frog World Frog

aaaaaaaaaaaaaaaaa

Fight Like A Brave Like A

Ebcdic Hell World Hell World

1World Hell Frog Hell Frog

Pullup World Hell

root@DESKTOP-5HM2HTK:~#

Спокойно склеит 3, 4, 5 и т.д. файлов.

Своего рода обратный cut.

С помощью ключа –d можно поменять разделитель на указанный:

root@DESKTOP-5HM2HTK:~# paste -d"^" file1 part2

Hell World Frog^World Frog

aaaaaaaaaaaaaaaaa^

Fight Like A Brave^Like A

Ebcdic Hell World ^Hell World

1World Hell Frog^Hell Frog

Pullup World Hell^

^

root@DESKTOP-5HM2HTK:~#

Ключ –s преобразует строчки каждого файла в одну строчку, а затем получившиеся строчки склеивает:

root@DESKTOP-5HM2HTK:~# paste -s file1 part2

Hell World Frog aaaaaaaaaaaaaaaaa Fight Like A Brave Ebcdic Hell World 1World Hell Frog Pullup World Hell

World Frog Like A Hell World Hell Frog

root@DESKTOP-5HM2HTK:~#

1. **uniq**

С помощью команды uniq можно удалять повторяющиеся подряд идущие строчки из файлов:

root@DESKTOP-5HM2HTK:~# cat file2

Hell Banana Ups

Ascii Potato

Ascii Potato

Hell Banana Ups

312312312

312312312

312312312

World Apple

Frog Microsoft

Unix Google

Well Banana Ops

root@DESKTOP-5HM2HTK:~# uniq file2

Hell Banana Ups

Ascii Potato

Hell Banana Ups

312312312

World Apple

Frog Microsoft

Unix Google

Well Banana Ops

root@DESKTOP-5HM2HTK:~#

Ключ –с показывает число повторений каждой из строчек:

root@DESKTOP-5HM2HTK:~# uniq -c file2

1 Hell Banana Ups

2 Ascii Potato

1 Hell Banana Ups

3 312312312

1 World Apple

1 Frog Microsoft

1 Unix Google

1 Well Banana Ops

root@DESKTOP-5HM2HTK:~#

С помощью ключа –d выводятся только повторяющиеся строки, -u – только неповторяющиеся:

root@DESKTOP-5HM2HTK:~# uniq -u file2

Hell Banana Ups

Hell Banana Ups

World Apple

Frog Microsoft

Unix Google

Well Banana Ops

root@DESKTOP-5HM2HTK:~# uniq -d file2

Ascii Potato

312312312

root@DESKTOP-5HM2HTK:~#

При ключе –i регистр не учитывается:

root@DESKTOP-5HM2HTK:~# uniq -c file2

1 Hell Banana Ups

2 Ascii Potato

1 Hell Banana Ups

3 312312312

1 World Apple

2 Frog Microsoft

1 frog microsoft

1 Unix Google

1 Well Banana Ops

root@DESKTOP-5HM2HTK:~# uniq -c -i file2

1 Hell Banana Ups

2 Ascii Potato

1 Hell Banana Ups

3 312312312

1 World Apple

3 Frog Microsoft

1 Unix Google

1 Well Banana Ops

root@DESKTOP-5HM2HTK:~#

С помощью ключа –f можно пропустить указанное количество слов в строчке:

root@DESKTOP-5HM2HTK:~# cat file3

Qwerty1 Qwerty2 Qwerty3

Frog Guap Qwerty3

Qwerty Zaza Qwerty3

Qwerty1 Qwerty2 qwerty3

root@DESKTOP-5HM2HTK:~# uniq -f 2 file3

Qwerty1 Qwerty2 Qwerty3

Qwerty1 Qwerty2 qwerty3

root@DESKTOP-5HM2HTK:~# uniq -i -f 2 file3

Qwerty1 Qwerty2 Qwerty3

root@DESKTOP-5HM2HTK:~#

1. **join**

И последняя команда – join. С её помощью можно объединять строки нескольких файлов в общее поле. Следует сказать, что файлы обязательно должны быть отсортированы (по номеру, то есть идти как 1,2,3 и т.д.).

Рассмотрим такую задачу. У нас есть три таблицы: таблица с номером предмета, а далее сам предмет; таблица с номером ученика и имя ученика; таблица с номером ученика, его номером предмета, по которому у него оценка, и сама оценка. Нам это всё надо перевести в человеческий вид:

root@DESKTOP-5HM2HTK:~# cat ycheniki

2 Петя

1 Настя

4 Коля

3 Лиза

root@DESKTOP-5HM2HTK:~# cat predmet

2 Физика

1 Биология

3 Математика

root@DESKTOP-5HM2HTK:~# cat ocenki

1 3 4

2 3 2

3 3 3

4 3 5

1 1 4

2 1 3

3 1 5

4 1 5

1 2 4

2 2 4

3 2 4

4 2 4

root@DESKTOP-5HM2HTK:~#

Сначала отсортируем таблицы, кроме главной, по первому столбцу. Например, сначала мы хотим сначала расшифровать номер предмета. Тогда сортируем нашу главную таблицу по второму столбцу и пишем следующее:

root@DESKTOP-5HM2HTK:~# cat ocenki.sort

1 1 4

2 1 3

3 1 5

4 1 5

1 2 4

2 2 4

3 2 4

4 2 4

1 3 4

2 3 2

3 3 3

4 3 5

root@DESKTOP-5HM2HTK:~# cat predmet.sort

1 Биология

2 Физика

3 Математика

root@DESKTOP-5HM2HTK:~# join -1 2 -2 1 ocenki.sort predmet.sort

1 1 4 Биология

1 2 3 Биология

1 3 5 Биология

1 4 5 Биология

2 1 4 Физика

2 2 4 Физика

2 3 4 Физика

2 4 4 Физика

3 1 4 Математика

3 2 2 Математика

3 3 3 Математика

3 4 5 Математика

root@DESKTOP-5HM2HTK:~# join -1 2 -2 1 ocenki.sort predmet.sort >ocenki.predmet

root@DESKTOP-5HM2HTK:~#

-1 – первый файл, 2 – столбец, по которому идёт соответствие

-2 – второй файл, 1 – столбец, по которому идёт соответствие

Теперь нам нужно расшифровать имена учеников. Упорядочим наш полученный файл по второй колонке и снова используем join:

root@DESKTOP-5HM2HTK:~# cat ocenki.predmet.sort

1 1 4 Биология

2 1 4 Физика

3 1 4 Математика

1 2 3 Биология

2 2 4 Физика

3 2 2 Математика

1 3 5 Биология

2 3 4 Физика

3 3 3 Математика

1 4 5 Биология

2 4 4 Физика

3 4 5 Математика

root@DESKTOP-5HM2HTK:~# cat ycheniki.sort

1 Настя

2 Петя

3 Лиза

4 Коля

root@DESKTOP-5HM2HTK:~# join -1 2 -2 1 ocenki.predmet.sort ycheniki.sort

1 1 4 Биология Настя

1 2 4 Физика Настя

1 3 4 Математика Настя

2 1 3 Биология Петя

2 2 4 Физика Петя

2 3 2 Математика Петя

3 1 5 Биология Лиза

3 2 4 Физика Лиза

3 3 3 Математика Лиза

4 1 5 Биология Коля

4 2 4 Физика Коля

4 3 5 Математика Коля

root@DESKTOP-5HM2HTK:~# join -1 2 -2 1 ocenki.predmet.sort ycheniki.sort >itog

root@DESKTOP-5HM2HTK:~#

Первые 3 колонки можно вырезать, так как они лишние:

root@DESKTOP-5HM2HTK:~# cut -f 3,4,5 -d" " itog >ITOG

root@DESKTOP-5HM2HTK:~# cat ITOG

4 Биология Настя

4 Физика Настя

4 Математика Настя

3 Биология Петя

4 Физика Петя

2 Математика Петя

5 Биология Лиза

4 Физика Лиза

3 Математика Лиза

5 Биология Коля

4 Физика Коля

5 Математика Коля

root@DESKTOP-5HM2HTK:~#

При желании, с помощью cut и paste колонки можно расположить в любом удобном порядке.

С практическим применением join разобрались. Теперь посмотрим его ключи.

Ключ –j заменяет запись “-1 … -2 …”, однако колонки должны быть в данном случае одинаковые:

root@DESKTOP-5HM2HTK:~# join -j 1 Ocenki Ycheniki

1 1 4 Настя

1 2 4 Настя

1 3 4 Настя

2 1 3 Петя

2 2 4 Петя

2 3 2 Петя

3 1 5 Лиза

3 2 4 Лиза

3 3 3 Лиза

4 1 5 Коля

4 2 4 Коля

4 3 5 Коля

root@DESKTOP-5HM2HTK:~# cat Ycheniki

1 Настя

2 Петя

3 Лиза

4 Коля

root@DESKTOP-5HM2HTK:~#

Ключ –a покажет, если вдруг при создании общего поля у какой то строки не нашлось пары:

2 после –a – указание номера файла

root@DESKTOP-5HM2HTK:~# cat Ycheniki

1 Настя

2 Петя

3 Лиза

4 Коля

5 Вася

root@DESKTOP-5HM2HTK:~#

root@DESKTOP-5HM2HTK:~# join -a 2 -j 1 Ocenki Ycheniki

1 1 4 Настя

1 2 4 Настя

1 3 4 Настя

2 1 3 Петя

2 2 4 Петя

2 3 2 Петя

3 1 5 Лиза

3 2 4 Лиза

3 3 3 Лиза

4 1 5 Коля

4 2 4 Коля

4 3 5 Коля

5 Вася

root@DESKTOP-5HM2HTK:~#

Используя ключ –v, напечатаются только строки, не имеющие пару:

root@DESKTOP-5HM2HTK:~# join -v 2 -j 1 Ocenki Ycheniki

5 Вася

root@DESKTOP-5HM2HTK:~#

При помощи ключа –t можно выбрать произвольный знак для разделения слов:

root@DESKTOP-5HM2HTK:~# join -t^ -j 1 Ocenki Ycheniki

1^1^4^Настя

1^2^4^Настя

1^3^4^Настя

2^1^3^Петя

2^2^4^Петя

2^3^2^Петя

3^1^5^Лиза

3^2^4^Лиза

3^3^3^Лиза

4^1^5^Коля

4^2^4^Коля

4^3^5^Коля

root@DESKTOP-5HM2HTK:~# cat Ycheniki

1^Настя

2^Петя

3^Лиза

4^Коля

5^Вася

root@DESKTOP-5HM2HTK:~#

Очевидно, что сопоставление колонок можно делать и по словам (не только по цифрам). Ключ –i позволяет игнорировать регистр:

root@DESKTOP-5HM2HTK:~# cat Ycheniki

five Вася

four Коля

one Петя

three Лиза

two Настя

root@DESKTOP-5HM2HTK:~# join -j 1 Ocenki Ycheniki

root@DESKTOP-5HM2HTK:~# join -i -j 1 Ocenki Ycheniki

Four 1 5 Коля

Four 2 4 Коля

Four 3 5 Коля

One 1 4 Петя

One 2 4 Петя

One 3 4 Петя

Three 1 5 Лиза

Three 2 4 Лиза

Three 3 3 Лиза

Two 1 3 Настя

Two 2 4 Настя

Two 3 2 Настя

root@DESKTOP-5HM2HTK:~#