

1 Команды для не подробного разбора

1.1 "cmp"

Команда "cmp" позволяет использовать утилиту для побайтового сравнения двух произвольных файлов. Она может использоваться как для сравнения бинарных, так и для сравнения текстовых файлов. В случае текстовых файлов для улучшения информативности вывода могут использоваться дополнительные параметры.

- -l — найти все различия в файлах и вывести их в виде списка
- -c — вывести не только номер отличающегося бита, но и его значение в обоих файлах
- -i — указать номера битов, с которых начать поиск различий

```
1 test> echo 'abc' > a
2 test> echo 'abcd' > b
3 test> cmp a b
4 a b differ: byte 4, line 1
5 test> echo 'abc' > b
6 test> cmp a b
7 test>
```

```
1 test> cat a
2 abcd
3 1234
4 5678
5 test> cat b
6 abc
7 1234
8 567
9 test> cmp -l a b
10 4 144 12
11 5 12 61
12 6 61 62
13 7 62 63
14 8 63 64
15 9 64 12
16 10 12 65
17 11 65 66
18 12 66 67
19 13 67 12
20 cmp: EOF on b after byte 13
21 test> cmp -lc a b
22 4 144 d 12 ^J
23 5 12 ^J 61 1
24 6 61 1 62 2
25 7 62 2 63 3
26 8 63 3 64 4
27 9 64 4 12 ^J
28 10 12 ^J 65 5
29 11 65 5 66 6
30 12 66 6 67 7
31 13 67 7 12 ^J
32 cmp: EOF on b after byte 13
33 test>
```

1.2 "tee"

Команда "tee" считывает стандартный ввод и записывает его одновременно в стандартный вывод и в один или несколько подготовленных файлов.

- -a — дополнить файл, а не перезаписать его

```
1 test> tee a b
2 this is "tee" command
3 this is "tee" command
4 test> cat a
5 this is "tee" command
6 test> cat b
7 this is "tee" command
8 test> cat a b
9 test>
```

```
1 test> cat a
2 this is "a" file
3 test> tee -a a
4 second string
5 second string
6 test> cat a
7 this is "a" file
8 second string
9 test>
```

1.3 "tr"

"tr" используется для замены, замещения или удаления символов из стандартного ввода, отправляя результат на стандартный вывод.

- -d — удалить знаки из набора
- -c — аменять последовательность знаков, которые повторяются, из перечисленных в наборе, на один такой знак

```
1 test> tr -d a
2 aaabbbababba
3 bbbbbb
4 test> tr -s ab
5 aaabbbbbbaaaabbbbb
6 abab
7 test>
```

1.4 "xxd", "od"

"xxd" — создаёт представление файла в виде шестнадцатеричных кодов или выполняет обратное преобразование. "od" — в виде восьмиричных кодов.

- -b — использовать вместо шестнадцатеричного кода биты
- -c — задаёт количество октетов, которое выводится на каждой строке
- -r — выполнить преобразование шестнадцатеричного кода в двоичный

- -u — использовать верхний регистр для шестнадцатиричных кодов

```

1 test> xxd -u -c 2 a
2 00000000: 7468 th
3 00000002: 6973 is
4 00000004: 2069 i
5 00000006: 7320 s
6 00000008: 2261 "a
7 0000000a: 2220 "
8 0000000c: 6669 fi
9 0000000e: 6C65 le
10 00000010: 0A73 .s
11 00000012: 6563 ec
12 00000014: 6F6E on
13 00000016: 6420 d
14 00000018: 7374 st
15 0000001a: 7269 ri
16 0000001c: 6E67 ng
17 0000001e: 0A .
18 test> od a
19 00000000 064164 071551 064440 020163 060442 020042 064546 062554
20 00000020 071412 061545 067157 020144 072163 064562 063556 000012
21 00000037
22 test>

```

1.5 "sum", "md5sum", "sha1sum", "sha256sum"

Все эти команды позволяют проверить контрольную сумму. Контрольная сумма — это цифра или строка, которая вычисляется путем суммирования всех цифр нужных данных. Ее можно использовать в дальнейшем для обнаружения ошибок в проверяемых данных при хранении или передаче. Тогда контрольная сумма пересчитывается еще раз и полученное значение сверяется с предыдущим.

```

1 test> sum a
2 29874 1
3 test> md5sum a
4 01740306f3b25b15c8bcc5959e524a62 a
5 test> sha1sum a
6 08ce3e6eb68bb5699639282aeadb5d4f7d69128 a
7 test> sha256sum a
8 6d40c3d4ce687c31420e3d79e762655e1c752c77f5a20ea12a47d4c9ecc0f8be a
9 test>

```

1.6 "file"

Команда "file" позволяет узнать тип данных, которые на самом деле содержатся внутри документа.

- -i — определение MIME-типа документа по его заголовку
- -l — список паттернов и их длина
- -z — анализ содержимого сжатых документов

```

1 test> file mt.jpg
2 mt.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment
   length 16, baseline, precision 8, 1080x846, components 3
3 test> file -i mt.jpg
4 mt.jpg: image/jpeg; charset=binary
5 test> file -l mt.jpg
6 Set 0:
7 Binary patterns:
8 Text patterns:
9 Set 1:
10 Binary patterns:
11 Text patterns:
12 Set 0:
13 Binary patterns:
14 Strength = 500@47: Biosig/Brainvision Marker file [biosig/brainvision]
15 Strength = 490@122: Biosig/TMSiLOG [biosig/tmsilog]
16 Strength = 461@127: Biosig/SYNERGY [biosig/synergy]
17 Strength = 460@46: Biosig/Brainvision V-Amp file []
18 ...

```

1.7 "split"

"split" позволяет разделить файл на несколько меньших файлов. Работает с любыми файлами, как текстовыми, так и бинарными.

- -b — задаёт размер выводимых файлов
- -n — задает количество выводимых файлов

```

1 test> ls -lh
2 total 136K
3 -rw-rw-r-- 1 matvey matvey 5,0K      25 11:38 int.txt
4 -rw-rw-r-- 1 matvey matvey 125K     25 11:26 mt.jpg
5 test> split int.txt -n 3
6 test> ls -lh
7 total 148K
8 -rw-rw-r-- 1 matvey matvey 5,0K      25 11:38 int.txt
9 -rw-rw-r-- 1 matvey matvey 125K     25 11:26 mt.jpg
10 -rw-rw-r-- 1 matvey matvey 1,7K      25 12:26 xaa
11 -rw-rw-r-- 1 matvey matvey 1,7K      25 12:26 xab
12 -rw-rw-r-- 1 matvey matvey 1,7K      25 12:26 xac
13 ...
14 test> split int.txt -b 1K
15 test> ls -lh
16 total 156K
17 -rw-rw-r-- 1 matvey matvey 5,0K      25 11:38 int.txt
18 -rw-rw-r-- 1 matvey matvey 125K     25 11:26 mt.jpg
19 -rw-rw-r-- 1 matvey matvey 1,0K      25 12:27 xaa
20 -rw-rw-r-- 1 matvey matvey 1,0K      25 12:27 xab
21 -rw-rw-r-- 1 matvey matvey 1,0K      25 12:27 xac
22 -rw-rw-r-- 1 matvey matvey 1,0K      25 12:27 xad
23 -rw-rw-r-- 1 matvey matvey 922       25 12:27 xae

```

1.8 "mktemp"

Команда "mktemp" создаёт временный файл. Временные файлы удаляются при каждом перезапуске системы

- `-d` — создать временную директорию

```

1 test> mktemp
2 /tmp/tmp.1aXN61jUIH
3 test> mktemp -d
4 /tmp/tmp.CxsrIzaVgs
5 test> cd /tmp/
6 tmp> ls -l
7 ...
8 -rw----- 1 matvey matvey    0          25 11:50 tmp.1aXN61jUIH
9 drwx----- 2 matvey matvey 4096          25 11:52 tmp.CxsrIzaVgs
10
11 ...

```

1.9 "comm"

Команда "comm" позволяет использовать одноименную утилиту для сравнения двух текстовых файлов с отсортированными по алфавиту строками. Перед использованием текстовых файлов строки в них должны быть отсортированы по алфавиту. При вызове без параметров рассматриваемая утилита будет выводить строки в трех столбцах: первый столбец будет содержать строки, присутствующие лишь в файле 1, второй столбец — строки, присутствующие лишь в файле 2, а третий столбец — строки, присутствующие в обоих файлах. Утилита поддерживает параметры `-1`, `-2` и `-3`, позволяющие не выводить столбцы под соответствующими номерами.

```

1 test> cat a
2 001
3 002
4 003
5 004
6 005
7 test> cat b
8 000
9 001
10 003
11 005
12 007
13 test> comm a b
14      000
15      001
16 001
17 002
18              003
19 004
20              005
21      007
22 test> comm -23 a b
23 001
24 002
25 004

```

1.10 "cut"

Команда "cut" используется, если нужно вырезать часть текста — при этом он может находиться в файле либо быть напечатанным через стандартный ввод.

- `-b` — указать номер или диапазон байтов для вырезания

- -c — указать символ или диапазон символов для вырезания
- -d — указать стандартный разделитель
- -f — указать перечень полей для вырезания

```

1 test> s="this.is.string"
2 test> echo \$s | cut -b 1
3 t
4 test> echo \$s | cut -f2 -d'. '
5 is
6 test> echo \$s | cut -c 1-13
7 this.is.strin
8 test> echo \$s | cut -c 13-26
9 ng

```

1.11 "paste"

"paste" позволяет объединять строки файлов по горизонтали. Она выводит строки, состоящие из последовательно соответствующих строк каждого файла, указанного в качестве аргумента, разделенных табуляцией.

- -d — указать стандартный разделитель
- -s — выводить строки последовательно, а не рядом друг с другом

```

1 test> cat a
2 abc
3 test> cat b
4 123
5 test> paste a b
6 abc      123
7 test> paste -d';' a b
8 abc;123
9 test>

```

1.12 "uniq"

Команда "uniq" предназначена для поиска одинаковых строк в массивах текста. При этом с найденными совпадениями пользователь может совершать множество действий — например, удалять их из вывода либо наоборот, выводить только их.

- -u — вывести только строки без повторов
- -d — если какая-нибудь строка повторяется несколько раз, то она будет выведена только один раз
- -D — выводить только повторяющиеся строки

```

1 test> cat a
2 abc
3 abc
4 123
5 abc
6 456
7 test> uniq a

```

```

8 abc
9 123
10 abc
11 456
12 test> uniq -c a
13     2 abc
14     1 123
15     1 abc
16     1 456
17 test> uniq -D a
18 abc
19 abc
20 test> uniq -d a
21 abc
22 test>

```

1.13 "grep"

Утилита "grep" выполняет поиск образца в текстовых файлах и выдает все строки, содержащие этот образец. Она использует компактный недетерминированный алгоритм сопоставления.

- -c — вывести только количество строк, содержащих образец
- -i — игнорировать регистр символов при сравнении
- -l — вывести только имена файлов, содержащих сопоставившиеся строки
- -n — вывести перед каждой строкой еще и её номер в файле
- -v — выдаёт все строки, кроме содержащих образец

```

1 test> grep "^LEFT" mt.sh
2 LEFT="$(cat "$2" | head -n1)"
3 test> grep -n "LEFT" mt.sh
4 16:LEFT="$(cat "$2" | head -n1)" #
5
6 36: if [[ "$LEFT" == '' ]]; then
7 41: CUR="${LEFT:${#LEFT}-1:1}"
8 42: LEFT="${LEFT:0:${#LEFT}-1}"
9 52: LEFT="$LEFT$TMP"
10 69: NEWRES="${1} '${LEFT}[$CUR]${RIGHT}'"
11 74: echo "'$(cat "$INPUT")' -> '$LEFT$CUR${RIGHT}'"
12 test> grep -c "LEFT$" mt.sh
13 0

```

1.14 "dd"

Утилита "dd" выполняет копирование данных из одного места в другое на двоичном уровне

- if — входной файл
- of — выходной файл
- bs — сколько байт читать и записывать за один раз
- cbs — сколько байт записывать за один раз

- count — скопировать указанное количество блоков, размер одного блока указывается в параметре bs
- ibs — читать указанное количество байт за раз

```

1 test> dd if=prog.mt of=out.mt bs=2 count=10
2 10+0 records in
3 10+0 records out
4 20 bytes copied, 0,000309346 s, 64,7 kB/s
5 test> cat out.mt
6 0, ,<,1
7 1,0,<,1
8 1,1,test>

```

1.15 "xargs"

"xargs" считывает данные из выходных данных другой команды или стандартного потока ввода (по умолчанию) и выполняет команду, переданную в качестве аргумента.

- -t — режим трассировки: команда и каждый построенный список аргументов перед выполнением выводится в стандартный поток ошибок.
- -p — режим с приглашением: xargs перед каждым вызовом команды запрашивает подтверждение. Включается режим трассировки (-t), за счет чего печатается вызов команды, который должен быть выполнен, а за ним — приглашение ?... Ответ у (за которым может идти что угодно) приводит к выполнению команды; при каком-либо другом ответе, включая возврат каретки, данный вызов команды игнорируется.

```

1 test> ls -lh
2 total 12K
3 -rw-rw-r-- 1 matvey matvey 6 25 15:16 input
4 -rwxrwxr-x 1 matvey matvey 3,3K 25 13:07 mt.sh
5 -rw-rw-r-- 1 matvey matvey 72 25 15:07 prog.mt
6 test> echo "file1 file2 file3" | xargs -t touch
7 touch file1 file2 file3
8 test> ls -lh
9 total 12K
10 -rw-rw-r-- 1 matvey matvey 0 25 15:21 file1
11 -rw-rw-r-- 1 matvey matvey 0 25 15:21 file2
12 -rw-rw-r-- 1 matvey matvey 0 25 15:21 file3
13 -rw-rw-r-- 1 matvey matvey 6 25 15:16 input
14 -rwxrwxr-x 1 matvey matvey 3,3K 25 13:07 mt.sh
15 -rw-rw-r-- 1 matvey matvey 72 25 15:07 prog.mt
16 test> echo "file1 file2 file3" | xargs -p rm
17 rm file1 file2 file3 ?...y
18 test> ls -lh
19 total 12K
20 -rw-rw-r-- 1 matvey matvey 6 25 15:16 input
21 -rwxrwxr-x 1 matvey matvey 3,3K 25 13:07 mt.sh
22 -rw-rw-r-- 1 matvey matvey 72 25 15:07 prog.mt
23 test>

```

1.16 "find"

"find" - это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

- -P — не открывать символические ссылки
- -L — получает информацию о файлах по символическим ссылкам
- -maxdepth — максимальная глубина поиска по подкаталогам
- -depth — искать сначала в текущем каталоге, а потом в подкаталогах
- -print — выводить полные имена файлов
- -type f — искать только файлы
- -type d — искать только директории

Искать можно по следующим категориям:

- -name — поиск по имени
- -perm — поиск по режиму доступа
- -user — поиск по владельцу
- -group — поиск по группе
- -mtime — поиск по времени модификации файла
- -atime — поиск по дате последнего чтения
- -newer — поиск файлов новее, чем указанный
- -size — поиск по указанному размеру

```

1 test> find / -maxdepth 2 -type f
2 /snap/README
3 /etc/crontab
4 /etc/ca-certificates.conf.dpkg-old
5 /etc/updatedb.conf
6 ...
7 ~> find -type d -perm 777
8 ./Downloads/fasm
9 ./Downloads/fasm/examples
10 ./Downloads/fasm/examples/elfobj
11 ...

```

1.17 "sort"

"sort" — это утилита для вывода текстовых строк в определенном порядке. Проще говоря, для сортировки. Ее можно использовать для сортировки текста из одного или нескольких файлов или с помощью нее может быть выполнена сортировка вывода для какой-либо команды.

- -b — не учитывать пробелы
- -n — сортировка строк по числовому значению
- -r — сортировать в обратном порядке
- -c — проверить был ли отсортирован файл

- -o — вывести результат в файл
- -k — указать поле, по которому нужно сортировать строки
- -a — указать символ-разделитель строк

```

1 test> sort -c prog.mt
2 sort: prog.mt:9: disorder: 2, ,=,2
3 test> sort prog.mt -o sort.mt
4 test> sort -c prog.mt
5 sort: prog.mt:9: disorder: 2, ,=,2
6 test> sort prog.mt -o sort.mt
7 test> cat sort.mt
8 0, ,<,1
9 1,0,<,1
10 1,1,<,1
11 1, ,>,2
12 2,0,1,3
13 2,1,0,3
14 2, ,=,2
15 3,0,>,2
16 3,1,>,2

```

1.18 "join"

"join"—объединяет строки двух файлов в общее поле. Для каждой пары строк ввода с одинаковыми общими полями, записывает строку в стандартный вывод. По умолчанию общее поле считается первым, поля разделяются пробельными знаками. Если один из ФАЙЛОВ задан как -, читает стандартный ввод.

- -e — замещать пустые строки указанной строкой
- -i — игнорировать регистр букв при сравнении полей
- -t — использовать заданный символ как разделитель поля ввода и поля вывода
- -1 — считать общим заданное поле из первого файла
- -2 — считать общим заданное поле из второго файла

```

1 test> cat a
2 001 aab
3 101 bab
4 111 bbb
5 test> cat b
6 001 —+
7 000 —
8 111 +++
9 test> sort b -o sb
10 test> sort a -o sa
11 test> cat sa
12 001 aab
13 101 bab
14 111 bbb
15 test> cat sb
16 000 —
17 001 —+
18 111 +++

```

```

19 test> join sa sb
20 001 aab —+
21 111 bbb +++

```

1.19 "df"

"df" — показывает список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования.

- -a — отобразить абсолютно все файловые системы
- -h — выводить размеры в читаемых форматах
- -H — выводить все размеры в гигабайтах
- -k — выводить размеры в килобайтах
- -total — выводить всю информацию про использование и доступное место
- -t — выводить информацию только про указанные файловые системы
- -x — выводить информацию обо всём, кроме указанных файловых систем

```

1 test> df -h
2 Filesystem      Size  Used Avail Use% Mounted on
3 udev            2,9G    0  2,9G   0% /dev
4 tmpfs           595M   1,9M  593M   1% /run
5 /dev/sda3       52G    33G   17G  67% /
6 tmpfs           3,0G   253M   2,7G   9% /dev/shm
7 tmpfs           5,0M    4,0K   5,0M   1% /run/lock
8 tmpfs           3,0G    0    3,0G   0% /sys/fs/cgroup
9 /dev/loop0      29M    29M    0 100% /snap/chromium-ffmpeg/15
10 /dev/loop1      99M    99M    0 100% /snap/core/10823
11 /dev/loop2      98M    98M    0 100% /snap/core/10583
12 /dev/loop3      33M    33M    0 100% /snap/chromium-ffmpeg/17
13 /dev/loop4      56M    56M    0 100% /snap/core18/1988
14 /dev/loop5      56M    56M    0 100% /snap/core18/1944
15 /dev/loop6      60M    60M    0 100% /snap/discord/120
16 /dev/loop7      76M    76M    0 100% /snap/discord/121
17 /dev/loop8     162M   162M    0 100% /snap/gnome-3-28-1804/128
18 /dev/loop9     218M   218M    0 100% /snap/gnome-3-34-1804/60
19 /dev/loop10     219M   219M    0 100% /snap/gnome-3-34-1804/66
20 /dev/loop11      65M    65M    0 100% /snap/gtk-common-themes/1514
21 /dev/loop12      32M    32M    0 100% /snap/snapd/10707
22 /dev/loop13      52M    52M    0 100% /snap/snap-store/518
23 /dev/loop14     163M   163M    0 100% /snap/gnome-3-28-1804/145
24 /dev/loop15      52M    52M    0 100% /snap/snap-store/498
25 /dev/loop16     2,3M   2,3M    0 100% /snap/gnome-system-monitor/157
26 /dev/loop17      32M    32M    0 100% /snap/snapd/11036
27 /dev/loop19     2,3M   2,3M    0 100% /snap/gnome-system-monitor/148
28 /dev/loop18      63M    63M    0 100% /snap/gtk-common-themes/1506
29 /dev/nvme0n1p2   96M    33M   64M  35% /boot/efi
30 tmpfs           595M    52K   595M   1% /run/user/1000
31 test> df -H -t ext4
32 Filesystem      Size  Used Avail Use% Mounted on
33 /dev/sda3       56G    35G   18G  67% /

```

1.20 "patch"

"patch" — программная утилита Unix, предназначенная для переноса правок (изменений) между разными версиями текстовых файлов. Информация о правке обычно содержится в отдельном файле, называемом «заплаткой», «правкой» или «файлом правки». Подобный файл, как правило, создается с помощью другой утилиты Unix — "diff", позволяющей автоматически извлечь информацию о различиях в тексте файлов.

```
1 test> cat a
2 001 aab
3 101 bab
4 111 bbb
5 test> cat b
6 001 —+
7 000 —
8 111 +++
9 test> diff a b > .patch
10 test> patch a .patch
11 patching file a
12 test> cat a
13 001 —+
14 000 —
15 111 +++
```

2 Команды для подробного разбора

2.1 "sed"

Команда "sed" это потоковый редактор текста, работающий по принципу замены. Его можно использовать для поиска, вставки, замены и удаления фрагментов в файле. С помощью этой утилиты вы можете редактировать файлы не открывая их. Будет намного быстрее если вы напишите что и на что надо заменить, чем вы будете открывать редактор, искать нужную строку и вручную всё заменять.

- -n — не выводить содержимое буфера шаблона в конце каждой итерации
- -e — команды, которые нужно выполнить для редактирования
- -f — проситать команды редактирования из файла
- -i — сделать резервную копию файла перед редактированием
- -r — включить расширенную поддержку синтаксиса регулярных выражений
- -s — если переданы несколько файлов, рассматривать их как разные потоки

У утилиты "sed" есть два буфера — активный буфер шаблона и дополнительный буфер. Оба изначально пусты. Программа выполняет заданные условия для каждой строки в переданном ей файле.

"sed" читает одну строку, удаляет из неё все завершающие символы и символы новой строки и помещает её в буфер шаблона. Затем выполняются переданные в параметрах команды, с каждой командой может быть связан адрес, это своего рода условие и команда выполняется только если подходит условие.

Когда всё команды будут выполнены и не указана опция -n, содержимое буфера шаблона выводится в стандартный поток вывода перед этим добавляется обратно символ перевода строки. если он был удален. Затем запускается новая итерация цикла для следующей строки.

Если не используются специальные команды, то после завершения одной итерации цикла содержимое буфера шаблона удаляется. Однако содержимое предыдущей строки хранится в дополнительном буфере и его можно использовать.

Каждой команде можно передать адрес, который будет указывать на строки, для которых она будет выполнена:

- **номер** — позволяет указать адрес строки, в которой надо выполнять команду
- **первая~шаг** — команда выполнится для первой указанной строки, а затем для всех с указанным шагом
- **\$** — последняя строка в файле
- **/regex/** — команда выполниться только для тех строк, которые подходят по регулярному выражения
- **номер, номер** — начиная с номера первой строки, и заканчивая номером последней
- **номер, +количество** — начиная с определённой строки и плюс ещё заданное количество строк после неё

Если для команды не был задан адрес, то она будет выполнена для всех строк.

Синтаксис регулярных выражений (все символы, кроме *, нужно экранировать):

- * — любое количество символов
- + — минимум один символ
- ? — один символ, либо отсутствие символа
- i — любые символы в количестве i
- i, j — любые символы в количестве от i до j
- i, — любые символы в количестве от i и больше

Также для использования "sed" нужно знать команды редактирования:

- q — завершает работу сценарий
- d — удаляет буфер шаблона и запускает следующую итерацию
- p — выводит содержимое буфера шаблона
- n — вывести содержимое буфера шаблона и просит в него следующую строку
- s — замена символов (с поддержкой регулярных выражений)
- y — замена символов (без поддержки регулярных выражений)
- w — записать содержимое буфера шаблона в файл
- N — добавить перевод строки к буферу шаблона
- g — заменить содержимое буфера шаблона содержимым дополнительного буфера

```
1 test> sed -n '21,24p' mt.sh
2 NEXT='n'
3 HELP='?'
4 RUN='r'
5
6 ...
7
8 test> cat a
9 this is a string; another string
10 this is a string too
11
12 test> sed 's/string/str/' a
13 this is a str; another string
14 this is a str too
15
16 test> sed 's/string/str/g' a
17 this is a str; another str
18 this is a str too
19
20 ...
21
22 test> cat prog.mt
23 0, ,<,1
24 1,0,<,1
25 1,1,<,1
```

```

26 1, ,>,2
27 2,0,1,3
28 2,1,0,3
29 3,0,>,2
30 3,1,>,2
31 2, ,=,2
32 test> sed 's/0/q0/g' prog.mt
33 q0, ,<,1
34 1,q0,<,1
35 1,1,<,1
36 1, ,>,2
37 2,q0,1,3
38 2,1,q0,3
39 3,q0,>,2
40 3,1,>,2
41 2, ,=,2
42
43 ...
44
45 test> find ~ -maxdepth 1 -type d | sed -e 's/\\///g' -e 's/homematvey/home/g'
46 home
47 homePictures
48 homeDownloads
49 home.conda
50 homefonts
51 home.mono
52 homeanaconda3
53 homeVideos
54 homesnap
55 home.vscode
56 homeMusic
57 home.pylint.d
58 homeDocuments
59 homecontest
60 home.mozilla
61 home.texlive2019
62 homeTemplates
63 home.yandex
64 home.local
65 home.emacs.d
66 home.pki
67 home.ssh
68 home.config
69 home.gnome
70 home.gnupg
71 home.vim
72 homePublic
73 homelabs
74 home.cache
75 home.dbus
76 homeDesktop
77 homedev
78 hometest
79 homescripts
80
81 ...
82
83 test> cat a
84 11
85 12

```

```

86 13
87 14
88 15
89 16
90 17
91 18
92 19
93 test> sed '1~1d' a
94 test> sed -n '1~3p' a
95 11
96 14
97 17
98 test>
99
100 ...
101
102 test> sed 5q a
103 11
104 12
105 13
106 14
107 15
108
109 ...
110
111 test> head a
112 total 2,1G
113 drwxr-xr-x  2 root root 4,0K      17 16:42 bin
114 drwxr-xr-x  4 root root 4,0K      24 08:55 boot
115 drwxrwxr-x  2 root root 4,0K       3 00:43 cdrom
116 drwxr-xr-x 21 root root 4,6K      25 17:49 dev
117 drwxr-xr-x 142 root root 12K      24 08:56 etc
118 drwxr-xr-x  3 root root 4,0K       3 00:44 home
119 lrwxrwxrwx  1 root root  32       3 00:46 initrd.img ->
    boot/initrd.img-5.4.0-42-generic
120 lrwxrwxrwx  1 root root  32       3 00:46 initrd.img.old ->
    boot/initrd.img-5.4.0-42-generic
121 drwxr-xr-x 22 root root 4,0K      16 14:58 lib
122 test> sed '/      /!d' a
123 drwxrwxr-x  2 root root 4,0K       3 00:43 cdrom
124 drwxr-xr-x  3 root root 4,0K       3 00:44 home
125 lrwxrwxrwx  1 root root  32       3 00:46 initrd.img ->
    boot/initrd.img-5.4.0-42-generic
126 lrwxrwxrwx  1 root root  32       3 00:46 initrd.img.old ->
    boot/initrd.img-5.4.0-42-generic
127 drwx-----  2 root root 16K       3 00:40 lost+found
128 drwxr-xr-x  3 root root 4,0K       3 00:51 media
129 drwxr-xr-x 13 root root 4,0K       3 17:15 snap
130 -rw-----  1 root root 2,0G       3 00:41 swapfile
131 lrwxrwxrwx  1 root root  29       3 00:46 vmlinuz -> boot/vmlinuz-5.4.0-42-generic
132
133 ...
134
135 test> wc a
136  29 260 1569 a
137 test> head a
138 total 2,1G
139 drwxr-xr-x  2 root root 4,0K      17 16:42 bin
140 drwxr-xr-x  4 root root 4,0K      24 08:55 boot
141 drwxrwxr-x  2 root root 4,0K       3 00:43 cdrom

```



```

142 drwxr-xr-x 21 root root 4,6K 25 17:49 dev
143 drwxr-xr-x 142 root root 12K 24 08:56 etc
144 drwxr-xr-x 3 root root 4,0K 3 00:44 home
145 lrwxrwxrwx 1 root root 32 3 00:46 initrd.img ->
    boot/initrd.img-5.4.0-42-generic
146 lrwxrwxrwx 1 root root 32 3 00:46 initrd.img.old ->
    boot/initrd.img-5.4.0-42-generic
147 drwxr-xr-x 22 root root 4,0K 16 14:58 lib
148 test> sed -i '/\| /d' a
149 test> wc a
150 15 128 721 a
151 test> head a
152 total 2,1G
153 drwxr-xr-x 21 root root 4,6K 25 17:49 dev
154 drwxr-xr-x 22 root root 4,0K 16 14:58 lib
155 drwxr-xr-x 2 root root 4,0K 16 14:58 lib32
156 drwxr-xr-x 2 root root 4,0K 16 14:58 lib64
157 -rw-r--r-- 1 root root 98 13 16:22 main.cpp
158 drwxr-xr-x 2 root root 4,0K 7 2020 mnt
159 drwxr-xr-x 5 root root 4,0K 28 2013 opt
160 dr-xr-xr-x 305 root root 0 24 21:51 proc
161 drwxr-xr-x 37 root root 960 25 16:08 run
162 test>
163
164 ...
165
166 test> sed = a | sed 'N;s/\n/\t/'
167 1 total 2,1G
168 2 drwxr-xr-x 21 root root 4,6K 25 17:49 dev
169 3 drwxr-xr-x 22 root root 4,0K 16 14:58 lib
170 4 drwxr-xr-x 2 root root 4,0K 16 14:58 lib32
171 5 drwxr-xr-x 2 root root 4,0K 16 14:58 lib64
172 6 -rw-r--r-- 1 root root 98 13 16:22 main.cpp
173 7 drwxr-xr-x 2 root root 4,0K 7 2020 mnt
174 8 drwxr-xr-x 5 root root 4,0K 28 2013 opt
175 9 dr-xr-xr-x 305 root root 0 24 21:51 proc
176 10 drwxr-xr-x 37 root root 960 25 16:08 run
177 11 drwxr-xr-x 2 root root 4,0K 7 2020 srv
178 12 dr-xr-xr-x 13 root root 0 24 21:51 sys
179 13 drwxrwxrwt 25 root root 4,0K 25 19:23 tmp
180 14 drwxr-xr-x 12 root root 4,0K 28 2013 usr
181 15 drwxr-xr-x 14 root root 4,0K 28 2013 var
182
183 ...
184
185 test> sed 's/[a-g]//g' a
186 totl 2,1G
187 rwxr-xr-x 21 root root 4,6K 25 17:49 v
188 rwxr-xr-x 22 root root 4,0K 16 14:58 li
189 rwxr-xr-x 2 root root 4,0K 16 14:58 li32
190 rwxr-xr-x 2 root root 4,0K 16 14:58 li64
191 -rw-r--r-- 1 root root 98 13 16:22 min.pp
192 rwxr-xr-x 2 root root 4,0K 7 2020 mnt
193 rwxr-xr-x 5 root root 4,0K 28 2013 opt
194 r-xr-xr-x 305 root root 0 24 21:51 pro
195 rwxr-xr-x 37 root root 960 25 16:08 run
196 rwxr-xr-x 2 root root 4,0K 7 2020 srv
197 r-xr-xr-x 13 root root 0 24 21:51 sys
198 rwxrwxrwt 25 root root 4,0K 25 19:23 tmp
199 rwxr-xr-x 12 root root 4,0K 28 2013 usr

```

```

200 | rwxr-xr-x  14 root root 4,0K          28  2013  vr
201 |
202 | ...
203 |
204 | test> sed 's/one/two/' b
205 | two
206 | ONE
207 | test> sed 's/one/two/I' b
208 | two
209 | two

```

2.2 "diff"

Команда "diff" предназначена для задеирования одноименной утилиты, осуществляющей сравнение двух указанных пользователем файлов и вывод информации об их различиях. Эта утилита может работать как с текстовыми, так и с бинарными файлами, но в подавляющем большинстве случаев используется для поиска различий в обычных текстовых файлах или файлах исходного кода. Благодаря существованию утилиты "patch" она может использоваться для внесения изменений в исходный код открытых программных проектов.

- -E — игнорировать изменения, связанные с добавлением символа табуляции в тексте
- -b — игнорировать изменения, связанные с добавлением пробелов
- -w — игнорировать изменения, связанные с добавлением пробелов и табуляции
- -B — игнорировать новые пустые строки
- -y — показать результаты в 2 колонки
- -r — просматривать каталоги рекурсивно
- -X — исключить из поиска файлы, названия которых совпадают с заданным шаблоном
- -d — исключать ложные срабатывания

```

1 | test> diff prog.mt sort.mt
2 | 6a7
3 | > 2, ,=,2
4 | 9d9
5 | < 2, ,=,2
6 |
7 | ...
8 |
9 | test> cat a
10 | #include <stdio.h>
11 | int main() {
12 |     printf("Hello World");
13 | }
14 | test> cat b
15 | #include <stdio.h>
16 | int main() {
17 |     int a;
18 |     scanf("%d", &a);
19 |     printf("%d", a);
20 |     printf("Hello World");
21 | }
22 | test> diff a b > patch.diff

```

```

23 test> patch a patch.diff
24 patching file a
25 test> cat a
26 #include <stdio.h>
27 int main(){
28     int a;
29     scanf("%d", &a);
30     printf("%d", a);
31     printf("Hello World");
32 }
33 test> diff a b
34
35 ...
36
37 test> cat a
38 aaa
39 bbb
40
41
42 test> cat b
43 aaa
44 bbb
45 test> diff a b
46 3,4d2
47 <
48 <
49 test> diff -E a b
50 3,4d2
51 <
52 <
53 test> diff -b a b
54 3,4d2
55 <
56 <
57 test> diff -B a b
58 test>
59
60 ...
61
62 test> cat a
63 first string
64 second string
65 third string
66
67 test> cat b
68 first str
69 second string
70 fourth string
71 test> diff -y a b
72 first string      | first str
73 second string     | second string
74 third string      | fourth string
75 test> diff -u a b
76 --- a    2021-02-25 19:37:58.049392694 +0300
77 +++ b    2021-02-25 19:38:11.693609826 +0300
78 @@ -1,4 +1,3 @@
79 -first string
80 +first str
81  second string
82 -third string

```

```

83 —
84 +fourth string
85
86 ...
87
88 labs> diff 20lab/ 21lab/
89 Only in 20lab/: 20lab.pdf
90 Only in 21lab/: 21lab.zip
91 Only in 21lab/: input
92 Only in 21lab/: mt.sh
93 Only in 21lab/: prog.mt
94 Only in 20lab/: test
95
96 ...
97
98 est> cat a
99 one
100 TWO
101 test> cat b
102 ONE
103 two
104 test> diff -u a b
105 — a    2021-02-25 19:48:06.802876730 +0300
106 +++ b    2021-02-25 19:48:15.479020096 +0300
107 @@ -1,2 +1,2 @@
108 -one
109 -TWO
110 +ONE
111 +two
112 test> diff -ui a b
113
114 ...
115
116 test> diff -q a b
117 Files a and b differ
118
119 ...
120
121 test> cat a
122 one
123 TWO
124 test> cat b
125 one
126 TWO
127 test> diff a b
128 test> diff -s a b
129 Files a and b are identical

```

2.3 "du"

Иногда возникает необходимость посмотреть сколько места занимают файлы в определённой папке и найти самые большие файлы для того чтобы их удалить. Конечно, для решения этих задач существует множество инструментов, но самый простой из них, это утилита "du". Она позволяет вывести размер всех файлов в определённой папке в байтах или в более удобном формате.

- -a — выводить размер для всех файлов, а не только для директорий (по умолчанию размер выводится только для директорий)
- -B — указать единицы вывода размера; доступны K, M, G, T, P, E, Z, Y

- -c — вывести в конце общий размер директорий
- -d — максимальная глубина вложенности директорий
- -h — выводить размер в единицах, удобных для чтения
- -L — следовать по всем символическим ссылкам
- -P — не следовать по символическим ссылкам (используется по умолчанию)
- -S — не включать размер поддиректорий в размер директории
- -si — выводить размер в СИ
- -s — вывести только общий размер
- -t — не учитывать файлы и директории размером меньше указанного
- -time — указывать время последней модификации файла
- -X — исключить файлы из подсчёта

```

1 ~> du ~/labs
2 12      /home/matvey/labs/6cp
3 20      /home/matvey/labs/21lab
4 12      /home/matvey/labs/20lab/test
5 168     /home/matvey/labs/20lab
6 204     /home/matvey/labs
7
8 ...
9
10 ~> du -c ~/labs
11 12      /home/matvey/labs/6cp
12 20      /home/matvey/labs/21lab
13 12      /home/matvey/labs/20lab/test
14 168     /home/matvey/labs/20lab
15 204     /home/matvey/labs
16 204     total
17
18 ...
19
20 ~> du -d 1 ~/labs
21 12      /home/matvey/labs/6cp
22 20      /home/matvey/labs/21lab
23 168     /home/matvey/labs/20lab
24 204     /home/matvey/labs
25
26 ...
27
28 ~> du -h ~/labs
29 12K     /home/matvey/labs/6cp
30 20K     /home/matvey/labs/21lab
31 12K     /home/matvey/labs/20lab/test
32 168K    /home/matvey/labs/20lab
33 204K    /home/matvey/labs
34
35 ...
36
37 ~> du -hS ~/labs
38 12K     /home/matvey/labs/6cp

```

```
39 20K      /home/matvey/labs/21lab
40 12K      /home/matvey/labs/20lab/test
41 156K     /home/matvey/labs/20lab
42 4,0K     /home/matvey/labs
43
44 ...
45
46 ~> du -si ~/labs
47 13k      /home/matvey/labs/6cp
48 21k      /home/matvey/labs/21lab
49 13k      /home/matvey/labs/20lab/test
50 173k     /home/matvey/labs/20lab
51 209k     /home/matvey/labs
52
53 ...
54
55 ~> du -s ~/labs
56 204      /home/matvey/labs
57
58 ...
59
60 ~> du -hst 1G ~
61 13G      /home/matvey
62
63 ...
64
65 ~> du -hst 1G --time ~
66 13G      2021-02-25 19:55      /home/matvey
```