

Linux

Работа с файлами

- Базовые команды по работе с файлами

echo cat more less head tail tee

sed awk cut tr

grep find

cmp diff patch

touch cp mv rm dd

basename dirname wc uniq sort

- Создание мягких / жестких ссылок

STDOUT – стандартный вывод

echo - выводит строку на стандартный вывод. В конец строки добавляет перенос на новую строку.

echo [OPTION]... [STRING]...

- n** не печатать завершающий символ новой строки
- e** обрабатывать спецсимволы (\t, \n, \\", и т.д.)

```
$ echo "Hello, world"
$ echo -n "Hello, "; echo world
$ echo -e "opa\nopa"
```

```
[root@localhost ~]# echo "Hello somebody!"
-bash: !": event not found
[root@localhost ~]# echo 'Hello somebody!'
Hello somebody!
[root@localhost ~]# echo -e "opa\nopa"
opa
opa
```

cat - утилита последовательно читает файлы и пишет их в стандартный вывод.

cat [OPTION]... [FILE]...

Аргументы FILE обрабатываются в порядке их следования в командной строке.

Если FILE задан как дефис или вовсе отсутствует, то cat производит чтение со стандартного ввода (STDIN).

В таком случае окончание ввода - Ctrl+D (^D)

- b** нумеровать непустые выводимые строки, начиная с 1
- n** нумеровать все выводимые строки, начиная с 1
- v** выводить непечатаемые символы в читабельном виде

```
$ cat file1 file2 > file3  
$ cat file1 - file2 - file3  
$ cat > file
```

less - быстрое и гибкое отображение, перемещение, поиск в больших файлах. Команды управления основаны на командах vi

less [options] files ...

-i, --ignore-case	регистронезависимый поиск (только для строчных букв)
-I, --IGNORE-CASE	регистронезависимый поиск
-J	отобразить столбец состояния поиска (слева)
-n, --line-numbers	не нумеровать строки
-N, --LINE-NUMBERS	нумеровать строки
-r, --raw-control-chars	обрабатывать все управляющие символы (raw)
-S, --chop-long-lines	«отсечение» длинных строк (без переноса)

```
$ less -S big_file
```

Опции можно указать и в процессе работы с less

Команды управления в less

q Q	выход
e j DownArrow	вперед на одну строку
y k UpArrow	назад на одну строку
b PgUp	назад на окно
Space PgDwn	вперед на окно
RightArrow	вправо на половину окна
LeftArrow	влево на половину окна
F	вперед с ожиданием (эквивалентно «tail -f»)
g <	в начало файла
G >	в конец файла
:e [file]	открыть файл file
:n	перейти к следующему файлу
= :f	отобразить информацию о файле
/pattern	прямой поиск по шаблону pattern
n (N)	следующее (предыдущее) вхождение при поиске

grep - поиск совпадений по шаблону в указанных файлах или стандартном вводе, и отображение результата поиска (строк)

grep [options] PATTERN [FILE...]

-A / -B / -C NUM	отобразить также и NUM строк после / до / до и после совпадения
--colour, --color	подсвечивать совпадения
-c, --count	отобразить только число совпадений (строк)
-H, --with-filename	отображать также имена файлов (не только совпадения)
-h, --no-filename	не отображать имена файлов (только совпадения)
-i, --ignore-case	регистронезависимый поиск
-l, --files-with-match	отображать только имена файлов в которых есть совпадения
-I	пропускать двоичные файлы (как если бы они не содержали совпадений)

Опции grep (продолжение)

-n, --line-number	отображать также и номера строк
-o, --only-matching	отображать только совпадения (не всю строку)
-R, -r, --recursive	обрабатывать файлы рекурсивно
-v, --invert-match	инверсия совпадений
-w, --word-regexp	только целые слова

```
> grep -rw 'root' /etc
/etc/fstab:/dev/mapper/centos-root /                xfs      defaults      0 0
/etc/grub.d/00_header:loadfont (\${root})`make_system_path_relative_to_its_root $x`
/etc/grub.d/00_header:set theme=(\${root})`make_system_path_relative_to_its_root $GRUB_THEME`
/etc/grub.d/01_users:    set superusers="root"
/etc/grub.d/01_users:    password_pbkdf2 root \${GRUB2_PASSWORD}
/etc/grub.d/10_linux:# loop-AES arranges things so that /dev/loop/X can be our root device, but
/etc/grub.d/10_linux:  \${linuxefi} \${rel_dirname}/${basename} root=\${linux_root_device_thisversion} ro \${arg
/etc/grub.d/10_linux:  linux\${sixteenbit} \${rel_dirname}/${basename} root=\${linux_root_device_thisversion} r
/etc/grub.d/20_linux_xen:# loop-AES arranges things so that /dev/loop/X can be our root device, but
/etc/grub.d/20_linux_xen:    module \${rel_dirname}/${basename} placeholder root=\${linux_root_device_thisv
/etc/grub.d/30_os-prober:    parttool \${root} hidden-
/etc/grub.d/30_os-prober:    drivemap -s (hd0) \${root}
/etc/grub.d/30_os-prober:    multiboot /boot/gnumach.gz root=device:\${mach_device}
/etc/grub.d/30_os-prober:    --exec-server-task='\${exec-task}' -T typed '\${root}' \\\
/etc/rc.d/init.d/network:  # Don't shut the network down if root or /usr is on NFS or a network
/etc/libnl/classid:ffff:ffff      root
/etc/group-:root:x:0:
/etc/default/grub:GRUB_CMDLINE_LINUX="rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet"
/etc/gshadow-:root:::
```


Регулярные выражения (regular expressions) – это система синтаксического разбора текстовых фрагментов по формализованному шаблону. Сам же шаблон (образец) задает правило поиска.

Базовые элементы регулярного выражения:

- Обычные символы (текст)

```
> grep ftp /etc/passwd  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

- Метасимволы (специальные символы)

[] \ ^ \$. | ? * + () { }

Нуждаются в экранировании при использовании в качестве текста:

Одиночное – «\»

Групповое – между «\Q» и «\E»

```
> grep ^ftp /etc/passwd  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

- Любой символ – «.» (кроме переноса на новую строку)

- Наборы символов (один из перечисленных символов)

Заключаются в «[», «]»: **[abcABC]**

Диапазон задается через дефис «-»: **[a-zA-C]**

Инверсия (не входит) – «^»: **[^0-9]**

```
> grep 'b[aou]sh' /etc/passwd
root:x:0:0:root:/root:/bin/bash
test:x:1000:1000:./home/test:/bin/bash
```

- Именованные классы (готовые наборы)

POSIX варианты:

[[:alnum:]], [[:alpha:]], [[:digit:]], [[:lower:]], [[:punct:]], [[:space:]], [[:upper:]], [[:xdigit:]]

Perl вариант (PCRE):

\d [0-9], десятичное число

(\D [^0-9], не десятичное число)

\w [a-zA-Z0-9_], символ слова

(\W - не символ слова)

\s [\t\n\r\f], пробельные символы

(\S - непробельный символ)

```
> grep -P '\d+' /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
> grep -E '[[[:digit:]]]+' /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

- Позиция в строке
 - ^ начало строки
 - \$ конец строки
 - \b граница слова
 - \B не граница слова
- Последовательности (квантификаторы, множители)
 - {n} ровно **n** повторений предыдущего шаблона
 - {n,m} от **n** до **m** повторений (оба включительно)
 - {n,} не менее **n** повторений
 - {,m} не более **m** повторений
 - * {0,}, ноль или более повторений
 - + {1,}, одно или более повторений
 - ? {0,1}, ноль или одно повторение
- Группировка: часть выражения между «(» и «)»
- Альтернативы (логическое «или»): разделение с помощью «|»

РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ (REGEXP)

```
> grep -horIP 'https?://[\w.]+' /etc | sort -u
http://0pointer.de
http://bugs.centos.org
http://debuginfo.centos.org
http://fedoraproject.org
http://gist.github.com
http://mirror.centos.org
http://mirrorlist.centos.org
http://proxy.yoyodyne.com
https://bugs.centos.org
https://bugzilla.redhat.com
http://stackoverflow.com
https://www.centos.org
https://www.maxmind.com
https://www.python.org
https://www.vaspects.com
http://vault.centos.org
http://wiki.xiph.org
http://www.deepspace6.net
http://www.domain.dom
http://www.freedesktop.org
http://www.gnu.org
http://www.iana.org
http://www.linuxpowertop.org
http://www.maxmind.com
http://www.pool.ntp.org
http://www.postfix.org
http://www.rsyslog.com
```

```
> grep -ri '^sip' /etc/services
sip          5060/tcp      # SIP
sip          5060/udp      # SIP
sip-tls      5061/tcp      # SIP-TLS
sip-tls      5061/udp      # SIP-TLS
```

```
> grep -vP '^$|^$*#' /etc/lvm/lvm.conf | head
config {
    checks = 1
    abort_on_errors = 0
    profile_dir = "/etc/lvm/profile"
}
devices {
    dir = "/dev"
    scan = [ "/dev" ]
    obtain_device_list_from_udev = 1
    external_device_info_source = "none"
```

```
> sed -r 's/(root|ftp)/\1_new/g' /etc/passwd |\
> grep _new
root_new:x:0:0:root_new:/root_new:/bin/bash
operator:x:11:0:operator:/root_new/sbin/nologin
ftp_new:x:14:50:FTP_User:/var/ftp_new/sbin/nologin
```

```
> sed -r 's!^([[alpha:]]+)[[:space:]]+([[digit:]]+)! \2 \1!' /var/log/messages
16 Apr 06:26:01 localhost rsyslogd: [origin software="rsyslogd" swVersion="8.24.0" x-pid="912"
16 Apr 06:37:05 localhost systemd: Got automount request for /proc/sys/fs/binfmt_misc, trigger
16 Apr 06:37:05 localhost systemd: Mounting Arbitrary Executable File Formats File System...
16 Apr 06:37:06 localhost systemd: Mounted Arbitrary Executable File Formats File System.
16 Apr 06:37:06 localhost kernel: ICMPv6: process `grep' is using deprecated sysctl (syscall)
```


find - утилита рекурсивно спускается по дереву каталогов, начиная с указанного, проверяя при этом каждый файл на соответствие указанным критериям

find путь... выражение

- name pattern** имя файла подпадает под шаблон
- iname pattern** подобен **-name**, но сравнение не учитывает регистр
- empty** файл или каталог пусты
- exec command \;** запуск команды `command` для каждого результата поиска
- inum num** номер индексного дескриптора (inode) файла равен **num**
- maxdepth n** погружение не более чем на **n** уровней вложенности каталогов
- print, -print0, -printf** различные способы вывода результатов поиска
- user user** пользователь-владелец файла с именем `user`
- group grp_name** группа-владелец файла с именем `grp_name`
- size n[cwbkMG]** файл использует **n** блоков по 512 байт, или:
 - b** → 512B, **c** → byte, **w** → 2B, **k** → KB, **M** → MB, **G** → GB
 - +n** — больше, чем **n**
 - n** — меньше, чем **n**

-type type - искомый файл имеет тип **type**:

b → блочное устройство, c → символьное устройство

d → каталог f → обычный файл

l → символьная ссылка p → именованный канал (FIFO)

s → сокет

-newer file искомый файл был изменен позже, чем file (время модификации)

-a... время доступа к файлу

-c... время изменения метаданных файла (права доступа и пр.)

-m... время изменения полезных данных файла

...min n ровно n минут назад с файлом что-то произошло (зависит от префикса)

Например, **-amin n** — ровно n минут назад было обращение к файлу

...time n ровно n суток назад с файлом что-то произошло (зависит от префикса).

В данном случае, **n** - кол-во полных суток ($n * 24h$), дробная часть **n** - игнорируется.

Например, **-atime 0** — доступ был менее суток назад;

-atime 1 — доступ был не менее суток назад, но не более двух.

+/-n — соответственно более/менее n суток назад.

```
> find /etc -type f -name "pass*"
/etc/pam.d/passwd
/etc/pam.d/password-auth-ac
/etc/passwd-
/etc/passwd
/etc/openssl/certs/password
```

```
> find -L /bin /sbin -iname "*fdisk"
/sbin/cfdisk
/sbin/fdisk
/sbin/sfdisk
```

```
> find . -type f -name "smth*" -exec rm -vf {} \;
removed './smth1'
removed './smth2'
removed './smth3'
removed './smth4'
removed './smth5'
```

```
> find . -type f -name "smth*" | xargs rm -vf
removed './smth1'
removed './smth2'
removed './smth3'
removed './smth4'
removed './smth5'
```

```
> find /var -type d -not -user root
/var/lib/postfix
/var/lib/chrony
/var/log/chrony
/var/spool/postfix/active
/var/spool/postfix/bounce
/var/spool/postfix/corrupt
/var/spool/postfix/defer
```

```
> find -L /sbin -inum 835
/sbin/mkfs.ext3
/sbin/mkfs.ext4
/sbin/mke2fs
/sbin/mkfs.ext2
```

```
> find /var -mmin -5
/var/lib/rpm
/var/lib/rpm/___db.001
/var/lib/rpm/___db.002
/var/lib/rpm/___db.003
/var/log/messages
/var/cache/yum/x86_64/7/base/mirrorlist.txt
/var/cache/yum/x86_64/7/timedhosts.txt
/var/cache/yum/x86_64/7/extras/mirrorlist.txt
/var/cache/yum/x86_64/7/updates/mirrorlist.txt
```

```
> find /var /usr -type f -size -30M -size +10M -exec du -h {} \;
22M    /var/cache/yum/x86_64/7/updates/gen/primary_db.sqlite
19M    /usr/lib/firmware/liquidio/lio_23xx_vsw.bin
```

sed - считывает указанные файлы или стандартный ввод, модифицирует их согласно указанным командам и выводит на стандартный вывод.

sed [OPTION]... {script-only-if-no-other-script} [input-file]...

- i[suffix]** модифицирует непосредственно сам файл;
если указан suffix — делает резервную копию файла
с таким суффиксом
- r** использовать расширенный REGEXP
- e command** дополнение списка команд
- n** не печатать исходные строки

Основные команды:

- s/<шаблон поиска>/<замена>/** поиск и замена (модификатор 'g' в конце —
замена в рамках всей строки, а не только первого нахождения)
- d** удалить строку
- p** вывести строку
- n[,m] command** учитывать строки только с n по m.
- /<pattern1>/,/<pattern2>/ command** работать со строками между результатами
поиска шаблонов

```
> grep -i --color system /etc/passwd
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
> sed -n 's/system/replaced/p' /etc/passwd
replacedd-network:x:192:192:systemd Network Management:/:/sbin/nologin
> sed -n 's/system/replaced/gp' /etc/passwd
replacedd-network:x:192:192:replacedd Network Management:/:/sbin/nologin
> sed -n 's/system/replaced/igp' /etc/passwd
replacedd-network:x:192:192:replacedd Network Management:/:/sbin/nologin
dbus:x:81:81:replaced message bus:/:/sbin/nologin
> █
```

```
> sed -r 's/(root|ftp)/\1_new/g' /etc/passwd | \
> grep _new
root_new:x:0:0:root_new:/root_new:/bin/bash
operator:x:11:0:operator:/root_new:/sbin/nologin
ftp_new:x:14:50:FTP User:/var/ftp_new:/sbin/nologin
```

```
> grep -i bootproto /etc/sysconfig/network-scripts/ifcfg-enp0s3
BOOTPROTO="dhcp"
> sed -ri.copy '/BOOTPROTO/ s/"./"/static/' /etc/sysconfig/network-scripts/ifcfg-enp0s3
> grep -i bootproto /etc/sysconfig/network-scripts/ifcfg-enp0s3
BOOTPROTO="static"
> ls /etc/sysconfig/network-scripts/ifcfg-enp0s3*
/etc/sysconfig/network-scripts/ifcfg-enp0s3  /etc/sysconfig/network-scripts/ifcfg-enp0s3.copy
```

awk - специальный язык для обработки текстов

Основное назначение:

- обработка текста;
- создание форматированного текстового вывода;
- выполнение арифметических действий;
- выполнение операций над строками и т.д.

Примеры:

```
$ awk '{print $3 "\t" $4}' input_file
```

вывести 3-ю и 4-ю колонку из файла input_file и разделить их в выводе символом табуляции

```
$ awk -F ':' '/^username/ {print $3}' /etc/passwd
```

вывести ID пользователя из файла /etc/passwd (-F задает разделитель колонок, /.../ - фильтр строк в виде регулярного выражения)

```
$ awk 'length($1) > 5' input_file
```

вывести все строки из файла input_file, в которых длина значения в 1-й колонке больше 5 знаков (если \$0 — всей строки).

cmp - сравнивает побайтно два файла и пишет результат в стандартный вывод. По умолчанию cmp не выдаёт никаких сообщений, если файлы одинаковы; если же они различаются, сообщается номер байта и строки, где обнаружено первое различие

cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]

SKIP1/SKIP2 пропустить указанное число байт в первом/втором файле

```
> cmp file1 file2
```

```
> cmp file1 file2 1 2
```

diff – сравнивает файлы построчно.

- i, --ignore-case** игнорировать разницу в регистре букв
- B** игнорировать разницу в пустых строках
- u, -U NUM, --unified[=NUM]** унифицированный формат, число строк в контексте
- r, --recursive** выполнять рекурсивное сравнение
- E, --ignore-tab-expansion** игнорировать изменения отступа
- N, --new-file** считать отсутствующие файлы пустыми

```
$ diff -u my.old my.new > my.patch  
$ diff -urBNE proj_dir.orig proj_dir > proj.patch
```

patch - утилита применяет патч (diff-файл), который был создан утилитой diff (одного из 4-х видов), на оригинал. При этом создается новая версия файла (возможно создание резервных копий).

- b** сохранять резервные копии
- dry-run** проверить патч, но не накладывать
- d dir, --directory=dir** предварительно перейти в указанную директорию
- E, --remove-empty-files** после наложения патчей удалить пустые файлы
- i patchfile** считать патч из указанного файла вместо stdin
- p[number], --strip[=number]** задать число элементов пути («/»), которое будет удалено в именах файлов внутри патча.
- R** откатить изменения патча

```
> patch < my.patch  
> patch -d x-proj < x-my.patch
```

```
> diff -u /etc/profile ~/profile new > ~/profile.patch  
> patch --verbose -p1 < /root/profile.patch  
Hmm... Looks like a unified diff to me...  
The text leading up to this was:  
-----  
|--- /etc/profile      2019-04-24 00:10:24.954572045 -0400  
|+++ /root/profile_new 2019-04-24 00:03:54.559851616 -0400  
-----  
patching file etc/profile  
Using Plan A...  
Hunk #1 succeeded at 57.  
done  
> █
```

cp – создание копий файлов и каталогов.

cp [-R] [-f | -i | -n] [-lpv] исходный_файл целевой_файл

cp [-R] [-f | -i | -n] [-lpv] исходный_файл ... целевой_каталог

Жесткие и символьные ссылки копируются как объекты, на которые они ссылаются. Целевой каталог должен существовать (если не указана опция -r).

- f** принудительная замена файлов (по возможности)
- i** запрос подтверждения замены файлов
- p** сохранять атрибуты исходных файлов при замене (по возможности)
- R, -r** рекурсивно (вместе со всеми вложениями, соблюдая иерархию)
- v** описывать выполняемые действия

mv — перемещение / переименование файлов (каталогов)

mv [-f | -i] [-v] источник цель

mv [-f | -i] [-v] источник ... каталог

- f** не запрашивать подтверждение перед перезаписью целевого пути
- i** запрос подтверждения на перезапись

```
> cp file1 file2 file3 dir
> cp -vR dir /tmp
> cp -fp file1 file2
```

```
> mv file1 file2
> mv -vf dir1 dir2
> mv -v file1 file2 file3 dir
```

head - утилита выводит указанное кол-во первых строк или байт каждого из указанных файлов либо стандартного ввода (если кол-во строк не указано - по умолчанию выводит 10)

head [option]... [file]...

-n num	(-num)	кол-во первых строк
-c num		кол-во первых байт

tail — выводит последнюю часть файла или стандартного ввода, на стандартный вывод.

tail [option]... [file]...

-n num	(-num)	кол-во последних строк
-n +num		кол-во последних строк, отступ от начала.
-c num		кол-во первых байт
-f		не выходить при достижении конца (ждать новых данных)
-F		аналогично -f, плюс проверка не был ли файл переименован
-q		Подавляет печать заголовков в случае, когда одновременно просматриваются несколько файлов

```
$ tail -Ff -n 0 /var/log/messages  
$ tail -n +20 file | tail -n 10
```

tee - утилита копирует стандартный ввод в стандартный вывод, помещая копию в ноль или более файлов

- a** добавлять вывод к файлам, а не перезаписывать их
- line-buffered** не использовать стандартный буфер, а выводить результат построчно, может вызвать уменьшение производительности

```
$ tail -Ff /var/log/messages | grep --line-buffered kernel | tee -a LOG
```

touch — утилита меняет время последнего доступа / время изменения полезных данных файла. Если целевой файл не существовал, то создает пустой файл.

touch [OPTION]... FILE...

- c** не создавать файл
- a** изменить время доступа к файлу
- m** изменить время модификации файла
- r file** сослаться на метки времени другого файла file
- t** установить время согласно модификатора [[CC]YY]MMDDhhmm[.SS]]

```
$ touch file  
$ touch -a file  
$ touch -r ref_file file
```


dd — утилита копирует файл, преобразуя и форматируя его согласно опций.

Стандартный размер блока - 512Б

bs=n	размер блока
count=n	копировать только n блоков
if=file	входной файл
of=file	выходной файл
seek=n	пропустить n блоков (размером obs) в выходном файле
skip=n	пропустить n блоков (размером ibs) от начала входного файла

```
> dd if=/dev/zero of=~ /zeros bs=1M count=100
> dd if=/dev/urandom of=/dev/sdb
> dd if=/dev/sda of=~ /partition.img bs=1 skip=446 count=64
```

```
> dd if=/dev/sda of=~ /mbr.copy bs=512 count=1
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.000167329 s, 3.1 MB/s
```

tr - преобразовать, сжать и/или удалить символы со стандартного ввода и записать результат в стандартный вывод. Если 1-я строка длиннее, то последний символ string2 дублируется пока не закончится string1.

tr [OPTION]... SET1 [SET2]

-d удалить символы встречающиеся в строке

```
> echo 'a black cat' | tr -d a  
blk ct
```

-s удалить повторяющиеся символы из строки (сжать)

```
> echo 'many   blank   spaces' | tr -s '  
many blank spaces
```

-c команда работает с символами, не входящими в указанное множество

```
> echo 'abc 123 ebd' | tr -dc '[:digit:]'  
123
```

```
> tr -cs "[:alpha:]" "\n" < file1  
> tr "[:lower:]" "[:upper:]" < file1  
> tr -cd "[:print:]" < file1  
> tr -sc '[A-z]' '\n' < /etc/fstab
```

cut — вырезает указанные аргументами список части из каждой строки каждого файла и пишет их в стандартный вывод

- b список** задаёт позиции в байтах
- c список** задаёт позиции в символах
- d разделитель** использовать указанный аргументом разделитель символ как разделитель полей вместо символа табуляции
- f список** поля, разделённые символом разделителя полей
- s** пропускать строки, в которых не встречается символ разделителя. (иначе такие строки выводятся в исходном виде)

```
> cut -d ':' -f 1,3 /etc/passwd  
> cut -d ' ' -f1-3,6- /var/log/messages
```

```
> cut -d "#" -f 1 /etc/services | tr -s '\n' | head  
tcpmux      1/tcp  
tcpmux      1/udp  
rje         5/tcp  
rje         5/udp  
echo        7/tcp  
echo        7/udp  
discard     9/tcp      sink null  
discard     9/udp      sink null  
systat      11/tcp      users  
> █
```

basename — от строки с путем к файлу оставляет только имя самого файла (опционально может обрезать суффикс с расширением файла)

basename NAME [SUFFIX]

```
basename /usr/bin/sort      -> "sort"
basename include/stdio.h .h -> "stdio"
basename -s .h include/stdio.h -> "stdio"
basename -a any/str1 any/str2 -> "str1" followed by "str2"
```

dirname — от строки с путем к файлу оставляет только путь без имени самого файла

dirname [OPTION] NAME...

```
dirname /usr/bin/          -> "/usr"
dirname dir1/str dir2/str  -> "dir1" followed by "dir2"
dirname stdio.h            -> "."
```

```
> dirname /etc/sysconfig/network-scripts/ifcfg-enp0s3
/etc/sysconfig/network-scripts
> basename /etc/sysconfig/network-scripts/ifcfg-enp0s3
ifcfg-enp0s3
```

WC - утилита пишет в стандартный вывод число строк, слов и байт, содержащихся в каждом файле, указанных в строке аргументов, либо прочитанных из стандартного ввода.

Стандартный порядок вывода имеет следующий формат:
строки, слова, байты и имя файла.

Действие команды по умолчанию равносильно указанию опций -l, -w и -c.

- c вывести кол-во байт
- l вывести кол-во строк
- m вывести кол-во символов
- w вывести кол-во слов

```
> grep -iwo word file | wc -l  
> wc -cm /etc/services
```

```
> cat /etc/issue  
\S  
Kernel \r on an \m  
  
> wc /etc/issue  
3  6 23 /etc/issue
```


sort – утилита сортировки строк текстовых данных.

Может работать как с файлами, так и со стандартным вводом.

-f, --ignore-case	игнорировать регистр символов
-M, --month-sort	сортировать как названия месяцев
-n, --numeric-sort	сортировать как числа
-r, --reverse	отобразить в обратном порядке
-k, --key=POS1[,POS2]	сортировать по ключу, перечню полей (столбцов)
-t, --field-separator=SEP	использовать SEP как разделитель полей (столбцов)
-u, --unique	выводить только уникальные записи

```
> grep -vP "^\$|^#" /etc/services | awk '{print $1}' | sort -u  
> sort -nt ':' -k 3 /etc/passwd
```

```
> du -s /etc/* | sort -nr | head -5  
22360  /etc/selinux  
7604   /etc/udev  
1004   /etc/pki  
656    /etc/services  
604    /etc/ssh
```

uniq - показать или фильтровать повторяющиеся строки

Утилита читает исходный файл (стандартный ввод), сравнивает соседние строки и если следующая строка идентична текущей — ее игнорирует, а если отличается — переходит на нее и далее производит новое сравнение.

Если одинаковые строки не являются соседними — они останутся и в выводе утилиты.

Если необходимо вывести уникальные строки глобально среди всех строк — предварительно данные необходимо отсортировать.

- c перед каждой строкой выводить число повторений этой строки
- d выводить только те строки, которые повторяются
- f N игнорировать при сравнении первые N полей (слов*) каждой строки
- s N игнорировать при сравнении первые N символов каждой строки
- u выводить только те строки, которые не повторяются
- i сравнивать строки без учёта регистра

```
$ uniq file.in file.out
```

```
$ uniq -ciu file.in > file.out2
```

Символьная ссылка (также Symbolic link, символическая ссылка) — специальный файл, для которого не формируются никакие данные, кроме одной текстовой строки с указанием пути к файлу, который должен быть открыт при попытке обратиться к данной ссылке.

Особенности символьных ссылок:

- Целью может быть любой объект (другая ссылка, обычный файл, каталог и пр.)
- может ссылаться на несуществующий файл (т.н. «битая» ссылка)
- может ссылаться на файлы других ФС

Создание символьной ссылки:

```
$ ln -s file symbolic_link
```

```
> ln -s some_file symb_link
> ln -s abc symb_link2
> ls -l
total 4
-rw-r--r-- 1 root root 7 Apr 19 17:57 some_file
lrwxrwxrwx 1 root root 9 Apr 19 18:00 symb_link -> some_file
lrwxrwxrwx 1 root root 3 Apr 19 18:02 symb_link2 -> 
> █
```

Жёсткая ссылка (hard link) в UFS-совместимых файловых системах — это возможность сослаться на файл без необходимости использовать отдельный файл для хранения пути.

Физически, жесткая ссылка представлена как запись в каталоге (directory entry), которая ассоциирует имя файла с номером конкретного индексного дескриптора (inode).

Особенности жестких ссылок:

- все жесткие ссылки — равнозначны;
- при создании файла создается первая жесткая ссылка;
- изменение метаданных файла возможно через любую из его жестких ссылок;
- метаданные файла содержат счетчик жестких ссылок;
- удаление файла (высвобождение памяти) возможно после удаления всех жестких ссылок на него (счетчик жестких ссылок при этом равен нулю);
- нельзя создать жесткую ссылку на файл из другой ФС, также как и на каталог.

Создание жесткой ссылки:

```
$ ln file hard_link
```

```
> ls -li /sbin/mkfs.ext*  
835 -rwxr-xr-x. 4 root root 96384 May 16 2018 /sbin/mkfs.ext2  
835 -rwxr-xr-x. 4 root root 96384 May 16 2018 /sbin/mkfs.ext3  
835 -rwxr-xr-x. 4 root root 96384 May 16 2018 /sbin/mkfs.ext4
```