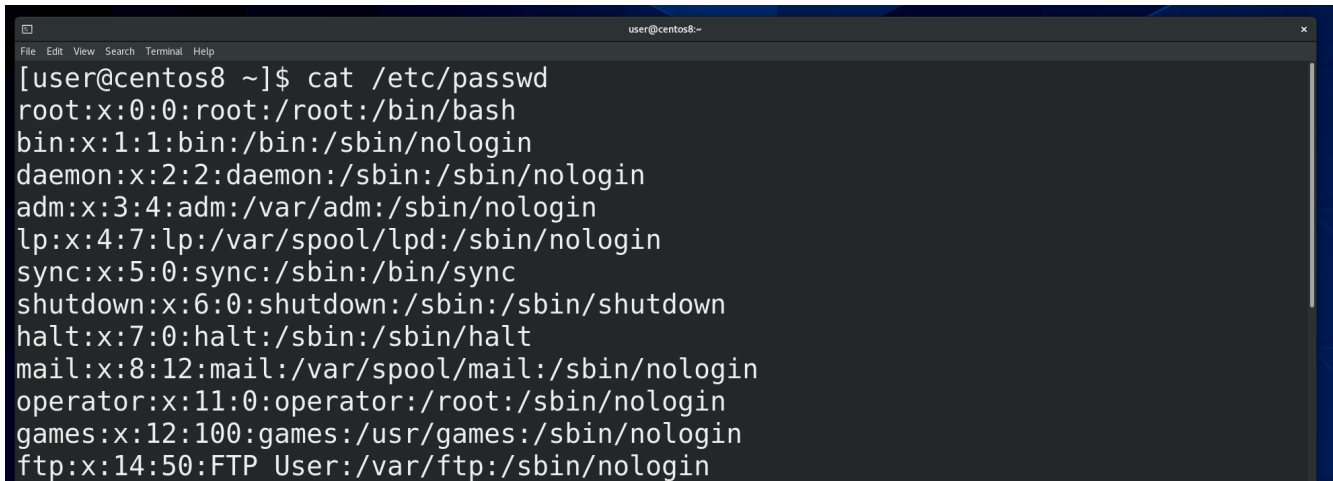
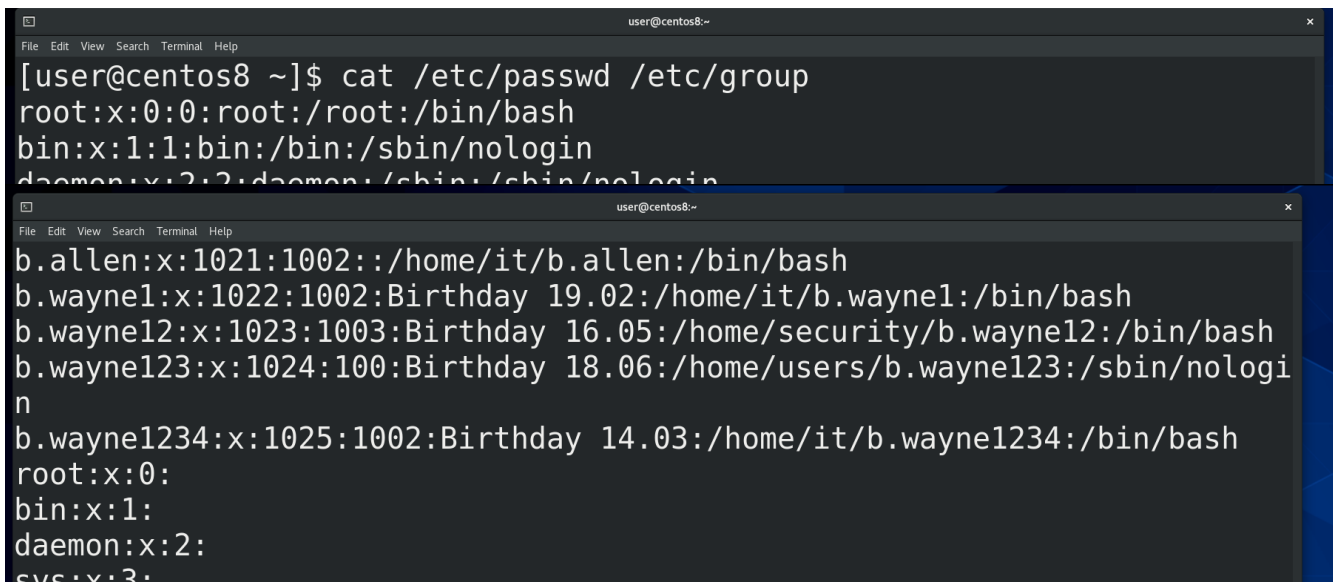


Как вы, возможно, поняли, в текстовом интерфейсе всё есть текст – команды, файлы, устройства - всё что угодно. В том числе это касается настроек и логов большинства программ, которые хранятся в виде текстовых файлов. Поэтому инструментов для работы с текстом на UNIX-подобных системах просто дофига. Какие-то из них очень простые, а о каких-то пишут книги на сотни страниц. Ну и чтобы научиться работать с текстом, нам нужен какой-то подопытный файл – и для этого идеально подойдёт файл `/etc/passwd` – файл, в котором хранится информация о пользователях в системе.

A terminal window titled 'user@centos8 ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'cat /etc/passwd' has been executed, displaying the contents of the file. The output lists system users and regular users in a colon-separated format.

```
[user@centos8 ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

Для начала посмотрим содержимое этого файла. С помощью команды `cat` мы можем вывести содержимое этого файла в терминал – `cat /etc/passwd`. Как видите, в терминале появилось много текста. Мы можем прокрутить его с помощью колёсика мыши, либо с помощью клавиш `shift+PgUp`, либо `shift+PgDn`. Запомните эти клавиши, потому что в виртуальном терминале мышка не работает.

Two terminal windows are shown. The top window shows the command 'cat /etc/passwd /etc/group' being executed, displaying the contents of both files. The bottom window shows a list of regular users from the /etc/passwd file, including b.allen, b.wayne1, b.wayne12, b.wayne123, and b.wayne1234, along with system users root, bin, daemon, and svcs.

```
[user@centos8 ~]$ cat /etc/passwd /etc/group
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
b.allen:x:1021:1002::/home/it/b.allen:/bin/bash
b.wayne1:x:1022:1002:Birthday 19.02:/home/it/b.wayne1:/bin/bash
b.wayne12:x:1023:1003:Birthday 16.05:/home/security/b.wayne12:/bin/bash
b.wayne123:x:1024:100:Birthday 18.06:/home/users/b.wayne123:/sbin/nologin
b.wayne1234:x:1025:1002:Birthday 14.03:/home/it/b.wayne1234:/bin/bash
root:x:0:
bin:x:1:
daemon:x:2:
svcs:x:3:
```

Команда `cat` подойдёт, когда у вас есть относительно небольшой текстовый файл и вам просто нужно посмотреть его содержимое. Но в целом `cat` – от слова конкатенация – может объединять вывод содержимого нескольких файлов. Для примера возьмём ещё один файл -

/etc/group – и выведем оба файла разом - cat /etc/passwd /etc/group. Если прокрутим вверх, то увидим - как только закончился один файл, начинается другой.

```
user@centos8 ~]$ cat -n /etc/passwd /etc/group
1 root:x:0:0:root:/root:/bin/bash
2 bin:x:1:1:bin:/bin:/sbin/nologin
3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
4 adm:x:3:4:adm:/var/adm:/sbin/nologin
5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
6 sync:x:5:0:sync:/sbin:/bin/sync
```

```
user@centos8 ~]$ cat -n /etc/passwd /etc/group
65 c.kent:x:1020:100::/home/users/c.kent:/sbin/nologin
66 b.allen:x:1021:1002::/home/it/b.allen:/bin/bash
67 b.wayne1:x:1022:1002:Birthday 19.02:/home/it/b.wayne1:/bin/bash
68 b.wayne12:x:1023:1003:Birthday 16.05:/home/security/b.wayne12:/bin/bash
69 b.wayne123:x:1024:100:Birthday 18.06:/home/users/b.wayne123:/sbin/nologin
70 b.wayne1234:x:1025:1002:Birthday 14.03:/home/it/b.wayne1234:/bin/bash
71 root:x:0:
72 bin:x:1:
73 daemon:x:2:
74 sys:x:3:
75 adm:x:4:
```

Ещё cat может пронумеровать строки с помощью ключа -n – cat -n /etc/passwd. Ну и обратите внимание, как эта опция работает при выводе двух файлов.

```
user@centos8 ~]$ tac /etc/passwd
b.wayne1234:x:1025:1002:Birthday 14.03:/home/it/b.wayne1234:/bin/bash
b.wayne123:x:1024:100:Birthday 18.06:/home/users/b.wayne123:/sbin/nologin
b.wayne12:x:1023:1003:Birthday 16.05:/home/security/b.wayne12:/bin/bash
b.wayne1:x:1022:1002:Birthday 19.02:/home/it/b.wayne1:/bin/bash
b.allen:x:1021:1002::/home/it/b.allen:/bin/bash
c.kent:x:1020:100::/home/users/c.kent:/sbin/nologin
o.queen:x:1019:1003::/home/security/o.queen:/bin/bash
b.wayne:x:1018:1002::/home/it/b.wayne:/bin/bash
```

У команды cat есть противоположная команда – tac, которая выводит текст реверсивно, то есть с конца – tac /etc/passwd.

```
user@centos8:~  
File Edit View Search Terminal Help  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
operator:x:11:0:operator:/root:/sbin/nologin  
games:x:12:100:games:/usr/games:/sbin/nologin  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin  
/etc/passwd
```

Как вы заметили, команда `cat` просто вывела содержимое файла на экран и всё. Если там сотни строк – придётся крутить вверх, вниз. Если у вас задача прочесть какой-то большой файл, то вам больше подойдёт программа-читалка, например, `less` – `less /etc/passwd`. Такие программы часто называют пейджерами. Стрелки, `Enter`, `pgup-pgdn`, пробел – с помощью всего этого можно листать.

```
user@centos8:~  
File Edit View Search Terminal Help  
  
SEARCHING  
  
/pattern      * Search forward for (N-th) matching line.  
?pattern      * Search backward for (N-th) matching line.  
n             * Repeat previous search (for N-th occurrence)  
.  
N             * Repeat previous search in reverse direction.  
ESC-n         * Repeat previous search, spanning files.  
ESC-N         * Repeat previous search, reverse dir. & span  
ing files.  
ESC-u         Undo (toggle) search highlighting.  
Spattern      * Display only matching lines
```

Если написать слэш (/) и текст, то `less` поищет этот текст в файле, а с помощью `n` или `N` можно перейти на следующее или предыдущее совпадение соответственно. Ну и `q` чтобы выйти. Похоже на `man`? А это потому что `man` использует `less` в качестве читалки. Если в `man` или в `less` нажать `h`, то откроется небольшой гайд по командам управления читалкой.

```
user@centos8:~  
[user@centos8 ~]$ head -4 /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
[user@centos8 ~]$ tail -3 /etc/passwd  
b.wayne12:x:1023:1003:Birthday 16.05:/home/security/b.wayne12:/bin/  
bash  
b.wayne123:x:1024:100:Birthday 18.06:/home/users/b.wayne123:/sbin/n  
ologin  
b.wayne1234:x:1025:1002:Birthday 14.03:/home/it/b.wayne1234:/bin/ba  
sh
```

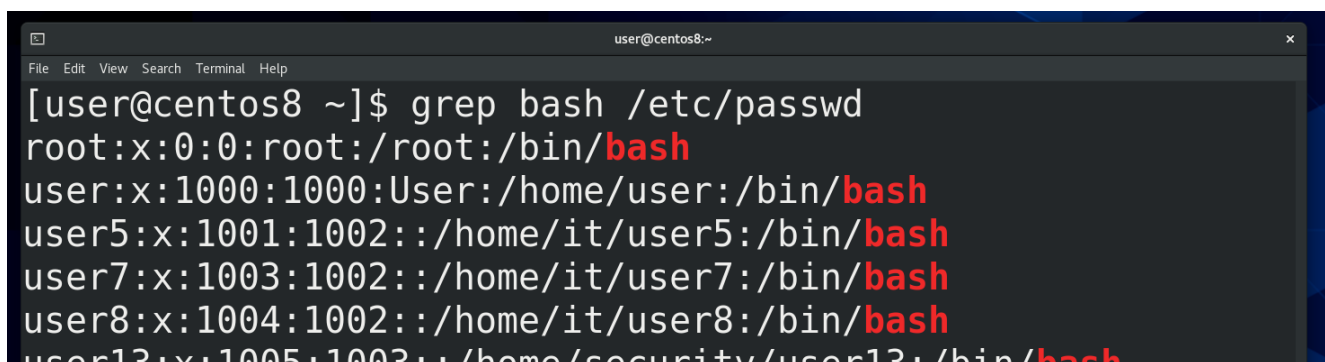
Но жизнь слишком коротка, чтобы читать большие файлы, поэтому часто пользуются двумя командами – head и tail – они показывают определённое количество строк с начала и с конца файла соответственно - head /etc/passwd или tail /etc/passwd. По умолчанию они показывают 10 строк, но можно указать что-то своё - head -4 /etc/passwd, tail -3 /etc/passwd. Ещё tail может читать с определённой строки, допустим всё что ниже 35 строки – tail -n +35 /etc/passwd.

```
user@centos8:~  
[user@centos8 ~]$ tail -f ~/.bash_history  
ls -s /usr/share/applications .  
ln -s /usr/share/applications .  
ls -l  
rm applications  
ln -s /usr/share/applications .  
ls -l  
ls  
touch  
command1  
command2  
command3
```

Одна из самых используемых опций tail – ключ f - может показывать, что добавляется в файл в реальном времени – tail -f logfile. Это часто применяется при решении проблем, когда вы видите кучу логов и вам нужно понять, что именно происходит при выполнении каких-то действий. Для примера посмотрим файл tail -f ~/.bash_history - сюда записываются выполненные

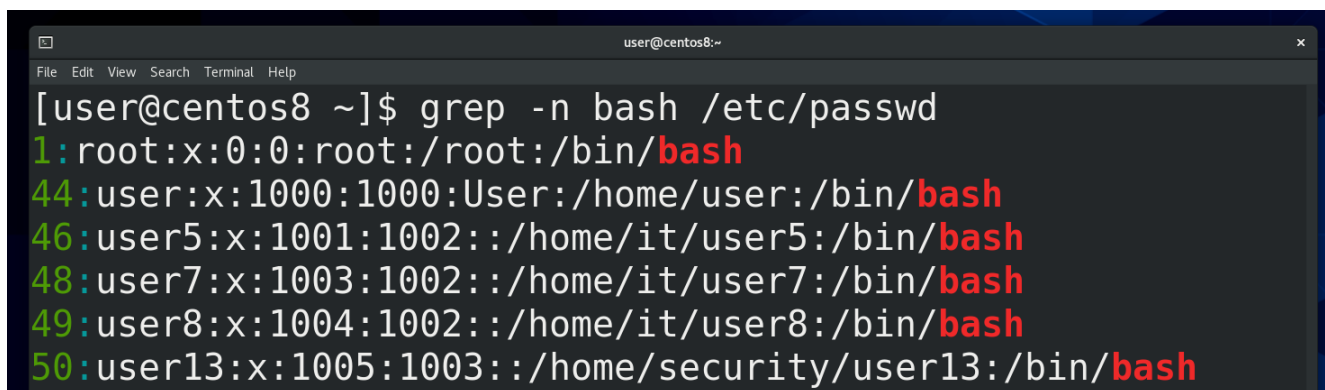
команды. Так вот, я открываю ещё один эмулятор терминала и ввожу какие-то команды. Потом закрываю новое окно и вижу, что у меня в этот файл добавились строчки. Когда текста много, не всегда понятно, что где куда добавилось, поэтому я могу зажать Enter и у меня появляется пустое пространство. Не беспокойтесь, это никак не влияет на сам файл, это просто для удобства чтения. Чтобы выйти, нажмите Ctrl+c. Возможно вам пока это не понятно, но просто запомните – если вам нужно открыть конец файла и ждать появления новых строчек, например в случае чтения логов, то используется tail с ключом f.

Вообще, одна из лучших фишек команд less, head и tail – то что они не пытаются прочесть весь файл целиком. Вы когда-нибудь пытались открыть текстовый файл на 100 мегабайт? А админы иногда сталкиваются с файлами с размером в пару гигабайт. Бывает - утром пришел на работу, а там сервер не работает. Смотришь – нет места на диске. Почему? А там лог файл на десяток или сотню гигабайт забил весь диск за одну ночь. Да, конечно, по хорошему такие ситуации легко предотвратить, но всё же речь о другом. И вот тебе нужно понять, что это там в логах такого на пару гигабайт. Сервера просто зависают при попытке открыть такие файлы. Но на линуксах есть эти утилиты, с помощью которых можно запросто прочитать файл любого размера. И так как, обычно, такие большие лог файлы забиваются одними и теми же строчками, то достаточно вывести, допустим, последние строк 50 файла, чтобы понять, что там произошло.



```
user@centos8:~  
[user@centos8 ~]$ grep bash /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
user:x:1000:1000:User:/home/user:/bin/bash  
user5:x:1001:1002::/home/it/user5:/bin/bash  
user7:x:1003:1002::/home/it/user7:/bin/bash  
user8:x:1004:1002::/home/it/user8:/bin/bash  
user13:x:1005:1003::/home/security/user13:/bin/bash
```

Ну и давайте напоследок затронем ещё одну команду – grep, хотя о ней мы ещё поговорим более подробно в другой раз. grep позволяет нам искать строки текста по шаблону. Допустим, в нашем файле /etc/passwd есть пользователи, которые пользуются интерпретатором bash. Я могу написать grep bash /etc/passwd и команда выдаст мне только строки, в которых есть слово bash.



```
user@centos8:~  
[user@centos8 ~]$ grep -n bash /etc/passwd  
1:root:x:0:0:root:/root:/bin/bash  
44:user:x:1000:1000:User:/home/user:/bin/bash  
46:user5:x:1001:1002::/home/it/user5:/bin/bash  
48:user7:x:1003:1002::/home/it/user7:/bin/bash  
49:user8:x:1004:1002::/home/it/user8:/bin/bash  
50:user13:x:1005:1003::/home/security/user13:/bin/bash
```

Я могу добавить опцию n - grep -n bash /etc/passwd – тогда я ещё увижу номера строк.

```
[user@centos8 ~]$ grep -rn bash /etc/
grep: /etc/crypttab: Permission denied
/etc/X11/xinit/xinitrc.d/10-qt5-check-opengl2.sh:1:#!/bin/b
ash
/etc/X11/xinit/Xclients:1:#!/bin/bash
/etc/X11/xinit/Xsession:1:#!/bin/bash
grep: /etc/libreport/cert-api.access.redhat.com.pem: Permis
sion denied
/etc/skel/.bash_logout:1:# ~/.bash_logout
/etc/skel/.bash_profile:1:# .bash_profile
```

grep может искать рекурсивно. То есть, я могу указать grep-у, чтобы он нашёл мне все упоминания bash в директории /etc – `grep -rn bash /etc/`. Как вы видите, вывода много, много где ошибки, потому что не хватает прав.

```
[user@centos8 ~]$ grep -rl bash /etc/
grep: /etc/crypttab: Permission denied
/etc/X11/xinit/xinitrc.d/10-qt5-check-opengl2.sh
/etc/X11/xinit/Xclients
/etc/X11/xinit/Xsession
grep: /etc/libreport/cert-api.access.redhat.com.pem: Permis
sion denied
/etc/skel/.bash_logout
/etc/skel/.bash_profile
/etc/skel/.bashrc
```

Я могу использовать ключ `l` – чтобы просто выводить имена файлов, в которых встречается слово bash – `grep -rl bash /etc/`.

```
[user@centos8 ~]$ grep -v bash /etc/passwd
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
```

Ну или я могу найти все строки, в которых не содержится это слово, то есть реверсировать поиск с помощью ключа `v` – `grep -v bash /etc/passwd`.

И вот мы затронули 4 команды для чтения текста – `cat`, `less`, `head` и `tail`, и даже научились искать текст с помощью команды `grep`. Осталось научиться писать.