

UNIX
BSD
AmigaOS
DOS
FreeDOS
IBM OS/2
ReactOS
Inferno OS
Plan 9
GNU/Linux
Microsoft Windows



Работа в оболочке UNIX/LINUX”

2021

Текстовые редакторы.

- emacs
- nano
- vim

Текстовый редактор nano.

```
GNU nano 2.3.1                               New Buffer                               Modified
getchar (); //Ожидание получения символа
for (i=1; i<=n; i++)
{
    fgets (arr, 100, file); //Чтение строки и запись в массив arr
    printf ("%s", arr); //вывод массива arr
    s--; //Уменьшение значения в счетчике
    fseek (file,k,SEEK_SET); //Перемещение метки на k-тую позицию
    k=k+35;
}
if (s<=0)
break; //Если s становится меньше нуля (строки заканчиваются),
// программа //заканчивает выполнение
}

fclose(file); //Закрытие файла
}

[ Read 39 lines ]

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit          ^J Justify     ^W Where Is    ^V Next Page    ^U UnCut Text  ^T To Spell
```

Быстрые клавиши nano.

- Символ каретки (^) - использование клавиши Ctrl.

Например, для перемещения вниз по странице нажмите и удерживайте клавишу Ctrl, а затем нажмите клавишу V.

Комбинация клавиш Ctrl + G служит для получения справки.

Быстрые клавиши nano.

- Команда M-\

M – это комбинация Esc, Alt . Символ \ просто представляет символ на клавиатуре.

Редактор по умолчанию.

Редактор по умолчанию : vim

Сменить умолчание:

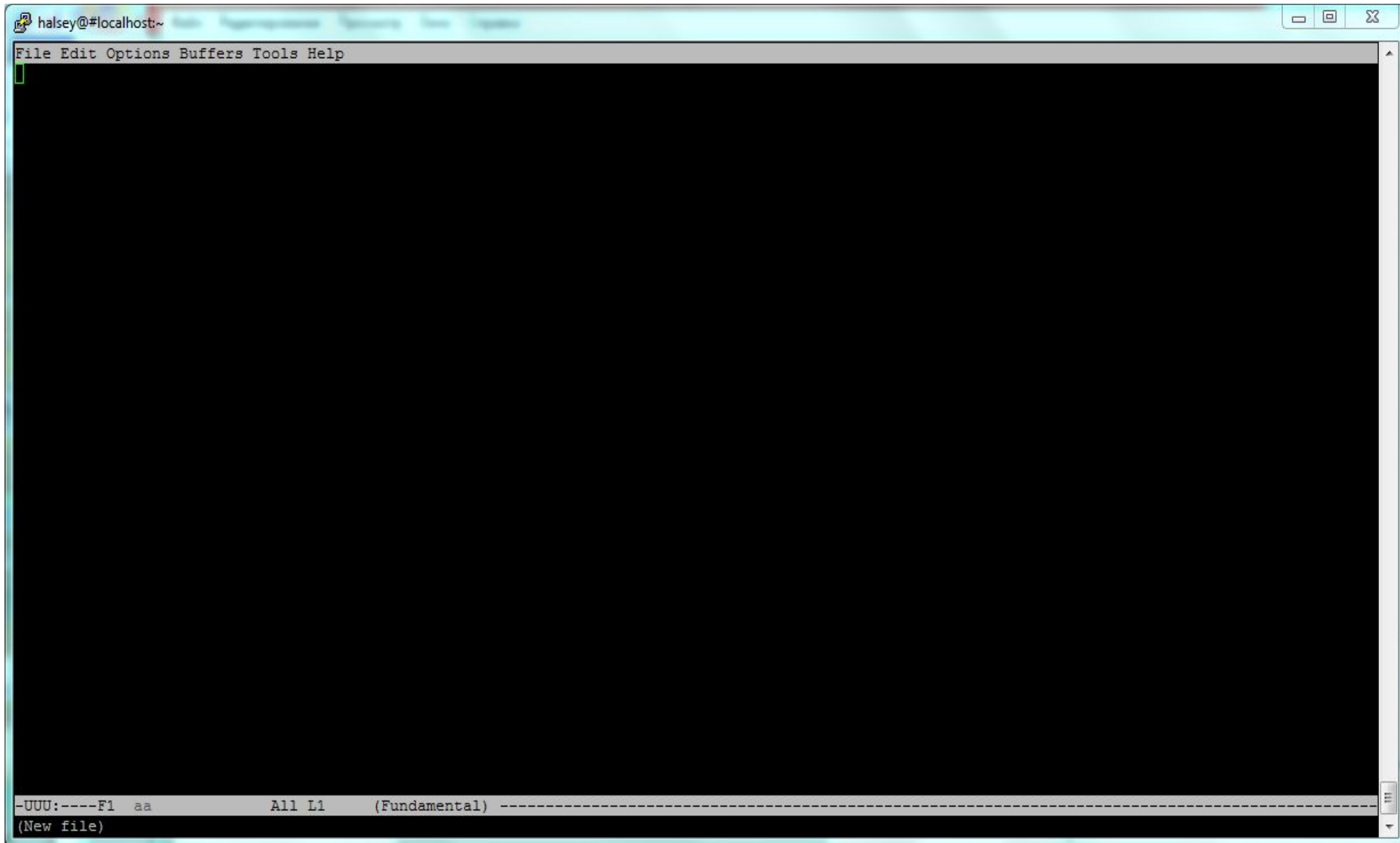
Для строковых редакторов(ed)

```
$ export EDITOR=ed
```

Для экранных редакторов(nano,emacs,vim)

```
$ export VISUAL=nano
```

Текстовый редактор emacs.



Примеры быстрых клавиш emacs

- Нажмите Ctrl + X, а затем комбинации клавиш Ctrl + S, чтобы сохранить содержимое буфера редактора в файл.
- Нажмите Ctrl + X, а затем комбинации клавиш Ctrl + C, чтобы выйти из редактора.
- Нажмите комбинацию клавиш Ctrl + H, а затем клавишу T, чтобы перейти к руководству по emacs .

Комбинации быстрых клавиш emacs

Комбинация клавиш:

C-<chr> - клавиша Ctrl и символ
клавиатуры

M-<chr> - клавиша Alt и символ клавиатуры

Редактор vi(vim)

```
$ which vim
```

```
/usr/bin/vim
```

```
$
```

```
$ which vi
```

```
alias vi='vim'
```

```
/usr/bin/vim
```

```
$
```

Режимы vim

Vim работает в буфере ОП.

Три стандартных режима :

1. Командный режим Это режим, который vim использует при первом входе в буферную область.

2. Режим вставки Режим вставки также называется режимом редактирования или ввода. Переход по клавише i . Выход - ESC

3. Ex Mode Этот режим команд двоеточия.

Перемещение по буферу vim.

Keystroke(s)	Description
h	Move cursor left one character.
l	Move cursor right one character.
j	Move cursor down one line (the next line in the text).
k	Move cursor up one line (the previous line in the text).

Перемещение по буферу vim.

Keystroke(s)	Description
w	Move cursor forward one word to front of next word.
e	Move cursor to end of current word.
b	Move cursor backward one word.
^	Move cursor to beginning of line.
\$	Move cursor to end of line.
gg	Move cursor to the file's first line.
G	Move cursor to the file's last line.
nG	Move cursor to file line number <i>n</i> .
Ctrl+B	Scroll up almost one full screen.
Ctrl+F	Scroll down almost one full screen.
Ctrl+U	Scroll up half of a screen.
Ctrl+D	Scroll down half of a screen.
Ctrl+Y	Scroll up one line.
Ctrl+E	Scroll down one line.

Команды редактирования текста в vim.

Keystroke(s)	Description
a	Insert text after cursor.
A	Insert text at end of text line.
dd	Delete current line.
dw	Delete current word.
i	Insert text before cursor.
I	Insert text before beginning of text line.
o	Open a new text line below cursor, and move to insert mode.
O	Open a new text line above cursor, and move to insert mode.
p	Paste copied text after cursor.
P	Paste copied (yanked) text before cursor.
yw	Yank (copy) current word.
yy	Yank (copy) current line.

Поиск в vim

В командном режиме:

1. Поиск вперед по буферу:

? Строка поиска

2. Поиск назад по буферу:

/ строка поиска

Общий синтаксис команд в командном режиме.

COMMAND [NUMBER-OF-TIMES] ITEM

Например:

Удаление 3 слов:

d 3 w

Скопировать текст от курсора до конца строки

y \$

Вставить текст от курсора

p

Команды режима Ex Mode .

Keystrokes	Description
<code>:! <i>command</i></code>	Execute shell <i>command</i> and display results, but don't quit editor.
<code>:r! <i>command</i></code>	Execute shell <i>command</i> and include the results in editor buffer area.
<code>:r <i>file</i></code>	Read <i>file</i> contents and include them in editor buffer area.

Команды режима Ex Mode (Сохранить).

Mode	Keystrokes	Description
Ex	:x	Write buffer to file and quit editor.
Ex	:wq	Write buffer to file and quit editor.
Ex	:wq!	Write buffer to file and quit editor (overrides protection).
Ex	:w	Write buffer to file and stay in editor.
Ex	:w!	Write buffer to file and stay in editor (overrides protection).
Ex	:q	Quit editor without writing buffer to file.
Ex	:q!	Quit editor without writing buffer to file (overrides protection).
Command	ZZ	Write buffer to file and quit editor.

Обработка текстовых данных

Конкатенация текстовых файлов:

```
$ cat numbers.txt random.txt
```

```
42
```

```
2A
```

```
52
```

```
0010 1010
```

```
*
```

```
42
```

```
Flat Land
```

```
Schrodinger's Cat
```

```
0010 1010
```

```
0000 0010
```

```
$
```

Обработка текстовых данных

Слияние текстовых файлов:

- `$ cat random.txt`
- 42
- Flat Land
- Schrodinger's Cat
- 0010 1010
- 0000 0010
- \$
- `$ cat numbers.txt`
- 42
- 2A
- 52
- 0010 1010
- *
- \$

Обработка текстовых данных

- Слияние файлов

```
$ paste random.txt numbers.txt
```

```
42 42
```

```
Flat Land 2A
```

```
Schrodinger's Cat 52
```

```
0010 1010 0010 1010
```

```
0000 0010 *
```

Обработка текстовых данных

- Команда трансформации содержимого файлов:

od [OPTION]... [FILE]...

- **\$ cat fortytwo.txt**
- 42
- forty two
- quarante deux
- zweiundvierzig
- forti to
- \$
- **\$ od fortytwo.txt**
- 0000000 031064 063012 072557 072162 020171 073564 005157 072561
- 0000020 071141 067141 062564 062040 072545 005170 073572 064545
- 0000040 067165 073144 062551 075162 063551 063012 071157 064564
- 0000060 072040 005157
- 0000064

Обработка текстовых данных

```
$ od -cb fortytwo.txt
```

```
0000000  4 2 \n f o u r t y t w o \n q u
```

```
064 062 012 146 157 165 162 164 171 040 164 167 157
```

```
012          161 165
```

```
0000020  a r a n t e d e u x \n z w e i
```

```
141 162 141 156 164 145 040 144 145 165 170 012 172
```

```
167 145 151
```

```
0000040  u n d v i e r z i g \n f o r t i
```

```
165 156 144 166 151 145 162 172 151 147 012 146 157
```

```
162 164 151
```

```
0000060  t o \n
```

```
040 164 157 012
```

```
0000064
```

```
$
```

Обработка текстовых данных

- Команда разделения файла:

```
split [ OPTION ]... [ INPUT [ PREFIX ]]
```


Пример разделения по кол-ву строк:

```
$ cat fortytwo.txt
```

- forty two
- quarante deux
- zweiundvierzig
- forti to

```
• $
```

```
• $ split -l 3 fortytwo.txt split42
```

```
• $
```

```
• $ ls split42*
```

```
• split42aa split42ab
```

```
• $
```

```
• $ cat split42aa
```

```
• 42
```

```
• forty two
```

```
• quarante deux
```

```
• $
```

```
• $ cat split42ab
```

```
• zweiundvierzig
```

```
• forti to
```

```
• $
```

Обработка текстовых данных- Команды форматирования файлов.

- Сортировка.

```
sort [ OPTION ]... [ FILE ]...
```

```
$ cat alphabet.txt
```

```
Alpha
```

```
Tango
```

```
Bravo
```

```
Echo
```

```
Foxtrot
```

```
$
```

```
$ sort alphabet.txt
```

```
Alpha
```

```
Bravo
```

```
Echo
```

```
Foxtrot
```

```
Tango
```

```
$
```

Обработка текстовых данных- Команды форматирования файлов.

- Сортировка числовых данных:
- **\$ sort counts.txt**
- 105
- 37
- 42
- 54
- 8
- **\$ sort -n counts.txt**
- 8
- 37
- 42
- 54
- 105
- **\$ sort -o newfile.sort -n counts.txt**

Обработка текстовых данных

- Нумерация строк в файле:

`nl [OPTION]... [FILE]...`

`$ nl ContainsBlankLines.txt`

- 1 Alpha
- 2 Tango

- 3 Bravo
- 4 Echo

- 5 Foxtrot

Обработка текстовых данных

- Нумерация строк в файле, учитывая пустые строки:

```
$ nl -ba ContainsBlankLines.txt
```

```
1 Alpha
```

```
2 Tango
```

```
3
```

```
4 Bravo
```

```
5 Echo
```

```
6
```

```
7
```

```
8 Foxtrot
```

```
$
```

Обработка текстовых данных.

- Утилиты просмотра содержимого:

1. `cat [OPTION] FILE [...]`
2. `more [OPTION] FILE [...]`
3. `Less [OPTION] FILE [...]`

Обработка текстовых данных.

- Утилиты просмотра содержимого:

```
head [ OPTION ]... [ FILE ]...
```

```
$ head /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
```

```
$
```

Обработка текстовых данных.

- Утилита head:
- **\$ head -n 2 /etc/passwd**
- root:x:0:0:root:/root:/bin/bash
- bin:x:1:1:bin:/bin:/sbin/nologin
- \$
- **\$ head -2 /etc/passwd**
- root:x:0:0:root:/root:/bin/bash
- bin:x:1:1:bin:/bin:/sbin/nologin
- \$

Обработка текстовых данных.

- Утилита tail :

```
tail [OPTION]... [FILE]...
```

```
$ tail /etc/passwd
```

```
saslauth:x:992:76:Saslauthd user:/run/saslauthd:/sbin/nologin
```

```
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
```

```
.....
```

```
tcpdump:x:72:72:::/sbin/nologin
```

```
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
```

Обработка текстовых данных.

```
$ tail -n 2 /etc/passwd
```

```
tcpdump:x:72:72:::/sbin/nologin
```

```
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
```

```
$
```

```
$ tail -n +42 /etc/passwd
```

```
gnome-initial-setup:x:989:983::/run/gnome-initial-setup:/sbin/nologin
```

```
tcpdump:x:72:72:::/sbin/nologin
```

```
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
```

```
$
```

Обработка текстовых данных.

- `$ sudo tail -f /var/log/auth.log`
- `[sudo] password for Christine:`
- `Aug 27 10:15:14 Ubuntu1804 sshd[15662]: Accepted password [...]`
- `Aug 27 10:15:14 Ubuntu1804 sshd[15662]: pam_unix(sshd:sess[...])`
- `Aug 27 10:15:14 Ubuntu1804 systemd-logind[588]: New sessio[...]`
- `Aug 27 10:15:50 Ubuntu1804 sudo: Christine : TTY=pts/1 ; P[...]`
- `Aug 27 10:15:50 Ubuntu1804 sudo: pam_unix(sudo:session): s[...]`
- `Aug 27 10:16:21 Ubuntu1804 login[10703]: pam_unix(login:se[...])`
- `Aug 27 10:16:21 Ubuntu1804 systemd-logind[588]: Removed se[...]`
- `^C`
- `$`

Обработка текстовых данных.

Некоторые дистрибутива LINUX имеют в своем составе демон journald. Просмотр журналов:

```
$ journalctl --follow <имя лог-файла>
```

Обработка текстовых данных.

- Команды обработки для получения сводной информации.

```
wc [ OPTION ]... [ FILE ]...
```

```
$ wc random.txt
```

```
5 9 52 random.txt
```

```
$
```

Обработка текстовых данных.

- Опции команды `wc`:

Short	Long	Description
<code>-c</code>	<code>--bytes</code>	Display the file's byte count.
<code>-L</code>	<code>--max-line-length</code>	Display the byte count of the file's longest line.
<code>-l</code>	<code>--lines</code>	Display the file's line count.
<code>-m</code>	<code>--chars</code>	Display the file's character count.
<code>-w</code>	<code>--words</code>	Display the file's word count.

Обработка текстовых данных.

- Утилита разделения содержимого на поля

Short	Long	Description
<code>-c nlist</code>	<code>--characters nlist</code>	Display only the record characters in the <i>nlist</i> (e.g., 1–5).
<code>-b blist</code>	<code>--bytes blist</code>	Display only the record bytes in the <i>blist</i> (e.g., 1–2).
<code>-d d</code>	<code>--delimiter d</code>	Designate the record's field delimiter as <i>d</i> . This overrides the Tab default delimiter. Put <i>d</i> within quotation marks to avoid unexpected results.
<code>-f flist</code>	<code>--fields flist</code>	Display only the record's fields denoted by <i>flist</i> (e.g., 1,3).
<code>-s</code>	<code>--only-delimited</code>	Display only records that contain the designated delimiter.
<code>-z</code>	<code>--zero-terminated</code>	Designate the record end-of-line character as the ASCII character NUL.

Обработка текстовых данных.

- Утилита обнаружения повторяющихся строк
- **\$ cat NonUniqueLines.txt**
- A
- C
- C
- A
- \$
- **\$ uniq NonUniqueLines.txt**
- A
- C
- A
- \$

Обработка текстовых данных.

- Утилита расчета хэш md5 для дайджестов сообщений.

```
$ md5sum fortytwo.txt
```

```
0ddaa12f06a2b7dcd469ad779b7c2a33
```

```
fortytwo.txt
```

```
$
```

Обработка текстовых данных.

- Другие алгоритмы хеширования
- `$ ls -l /usr/bin/sha???sum`
- `/usr/bin/sha224sum`
- `/usr/bin/sha256sum`
- `/usr/bin/sha384sum`
- `/usr/bin/sha512sum`
- `$`

Обработка текстовых данных.

```
$ sha256sum fortytwo.txt
```

```
0b2b6e2d8eab41e73baf0961ec707ef98978bcd8c7  
74ba8d32d3784aed4d286b fortytwo.txt
```

```
$
```

```
$ sha512sum fortytwo.txt
```

```
ac72599025322643e0e56cff41bb6e22ca4fbb76b1d  
7fac1b15a16085edad65ef55bbc733b8b68367723ced  
3b080dbaedb7669197a51b3b6a31db814802e2f31  
fortytwo.txt
```

```
$
```

Обработка текстовых данных.

- Команды использующие регулярные выражения:

```
grep [OPTION] PATTERN [FILE...]
```

```
$ grep root /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
```

```
operator:x:11:0:operator:/root:/sbin/nologin
```

```
$
```

Обработка текстовых данных.

- Опции grep:

Short	Long	Description
-c	--count	Display a count of text file records that contain a <i>PATTERN</i> match.
-d action	--directories=action	When a file is a directory, if <i>action</i> is set to read, read the directory as if it were a regular text file; if <i>action</i> is set to skip, ignore the directory; and if <i>action</i> is set to recurse, act as if the -R, -r, or --recursive option was used.
-E	--extended-regexp	Designate the <i>PATTERN</i> as an extended regular expression.
-i	--ignore-case	Ignore the case in the <i>PATTERN</i> as well as in any text file records.
-R, -r	--recursive	Search a directory's contents, and for any subdirectory within the original directory tree, consecutively search its contents as well (recursively).
-v	--invert-match	Display only text files records that do <i>not</i> contain a <i>PATTERN</i> match.

Обработка текстовых данных.

- `$ cat accounts.txt`
- sshd
- Christine
- nfsnobody
- `$`
- `$ fgrep -f accounts.txt /etc/passwd`
- sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
- Christine:x:1001:1001::/home/Christine:/bin/bash
- nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
- `$`
- `$ grep -F -f accounts.txt /etc/passwd`
- sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
- Christine:x:1001:1001::/home/Christine:/bin/bash
- nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
- `$`

Регулярные выражения.

- Basic regular expressions (BRE):

. * - несколько любых символов;

. - один любой символ;

[a,e,i,o,u] - список символов;

[A-z] - диапазон символов;

Якоря:

^ - в начале строки;

\$ - в конце строки

Регулярные выражения.

- **\$ grep daemon.*nologin /etc/passwd**
- daemon:x:2:2:daemon:/sbin:/sbin/nologin
- [...]
- daemon:/dev/null:/sbin/nologin
- [...]
- \$
- **\$ grep root /etc/passwd**
- root:x:0:0:root:/root:/bin/bash
- operator:x:11:0:operator:/root:/sbin/nologin
- \$
- **\$ grep ^root /etc/passwd**
- root:x:0:0:root:/root:/bin/bash
- \$

Регулярные выражения.

- `$ grep -v nologin$ /etc/passwd`
- `root:x:0:0:root:/root:/bin/bash`
- `sync:x:5:0:sync:/sbin:/bin/sync`
- `[...]`
- `Christine:x:1001:1001:~/home/Christine:/bin/bash`
- `$`

Регулярные выражения.

- Не выводить(игнорировать) пустые строки файла:

```
$ grep -v ^$ filename
```

Регулярные выражения.

- Специальная группа bracket expressions - классы символов(*character classes*):

`[:alnum:]` Matches any alphanumeric characters (any case), and is equal to using the `[0-9A-Za-z]` bracket expression

`[:alpha:]` Matches any alphabetic characters (any case), and is equal to using the `[A-Za-z]` bracket expression

`[:blank:]` Matches any blank characters, such as tab and space

`[:digit:]` Matches any numeric characters, and is equal to using the `[0-9]` bracket expression

`[:lower:]` Matches any lowercase alphabetic characters, and is equal to using the `[a-z]` bracket expression

Регулярные выражения.

- Специальная группа bracket expressions - классы символов (*character classes*):

[`:punct:`] Matches punctuation characters, such as `!`, `#`, `$`, and `@`

[`:space:`] Matches space characters, such as tab, form feed, and space

[`:upper:`] Matches any uppercase alphabetic characters, and is equal to using the

[`A-Z`] bracket expression

Регулярные выражения.

- Примеры использования классов символов:
- `$ cat random.txt`
- 42
- Flat Land
- Schrodinger's Cat
- 0010 1010
- 0000 0010
- \$
- `$ grep [[:digit:]] random.txt`
- 42
- 0010 1010
- 0000 0010
- \$

Расширенные регулярные выражения(ERE).

```
$ grep -E "^root|^dbus" /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
```

```
dbus:x:81:81:System message bus:/:/sbin/nologin
```

```
$
```

```
$ egrep "(daemon|s).*nologin" /etc/passwd
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
[...]
```

Опция -E тоже самое, что и egrep. То что в () – это подвыражение.

Потоки, перенаправления и каналы.

- Перенаправление:

Каждый объект обрабатывается как файл;

Каждый файл имеет файловый дескриптор;

0 – stdin, 1 – stdout, 2 – stderr

Примеры:

```
$ rm example.txt 2>1.txt
```

```
$ rm example.txt >1.txt 2>&1
```

По умолчанию stdout и stderr - /dev/tty

Потоки, перенаправления и каналы.

- Перенаправление:

```
$ echo "Nov 16, 2019" > AccountAudit.txt
```

```
$
```

```
$ wc -l /etc/passwd >> AccountAudit.txt
```

```
$
```

```
$ cat AccountAudit.txt
```

```
Nov 16, 2019
```

```
44 /etc/passwd
```

```
$
```


Потоки, перенаправления и каналы.

- Перенаправление стандартного потока ошибок:
- **\$ grep -d skip hosts: /etc/***
- grep: /etc/anacrontab: Permission denied
- grep: /etc/audisp: Permission denied
- [...]
- \$
- **\$ grep -d skip hosts: /etc/* 2> err.txt**
- /etc/nsswitch.conf:#hosts: db files nisplus nis dns
- /etc/nsswitch.conf:hosts: files dns myhostname
- [...]
- \$
- **\$ cat err.txt**
- grep: /etc/anacrontab: Permission denied
- grep: /etc/audisp: Permission denied
- [...]

Потоки, перенаправления и каналы.

- `$ grep -d skip hosts: /etc/* 2> /dev/null`
- `/etc/nsswitch.conf:#hosts: db files nisplus nis
dns`
- `/etc/nsswitch.conf:hosts: files dns
myhostname`
- `[...]`
- `$`

COMMAND1 | COMMAND2 [| COMMANDN]...

- Перенаправление ввода.

- **\$ cat Grades.txt**

- 89 76 100 92 68 84 73

- \$

- **\$ tr " " ", " < Grades.txt**

- 89,76,100,92,68,84,73

- \$

Резюме по перенаправлению ПОТОКОВ.

- > Redirect STDOUT to specified file. If file exists, overwrite it. If it does not exist, create it.
- >> Redirect STDOUT to specified file. If file exists, append to it. If it does not exist, create it.
- 2> Redirect STDERR to specified file. If file exists, overwrite it. If it does not exist, create it.
- 2>> Redirect STDERR to specified file. If file exists, append to it. If it does not exist, create it.
- &> Redirect STDOUT and STDERR to specified file. If file exists, overwrite it. If it does not exist, create it.
- &>> Redirect STDOUT and STDERR to specified file. If file exists, append to it. If it does not exist, create it.
- < Redirect STDIN from specified file into command.
- <> Redirect STDIN from specified file into command and redirect STDOUT to specified file.

Использование pipe(канала).

- Синтаксис :

COMMAND1 | COMMAND2 [| COMMANDN]...

```
$ grep /bin/bash$ /etc/passwd | wc -l
```

```
3
```

```
$
```

```
$ grep /sbin/nologin$ /etc/passwd | cut -d ":" -f 1 | sort | less
```

```
abrt
```

```
adm
```

```
avahi
```

```
bin
```

```
chrony
```

```
[...]
```

```
:
```

Использование утилиты tee

- **\$ grep /bin/bash\$ /etc/passwd | tee BashUsers.txt**
- root:x:0:0:root:/root:/bin/bash
- user1:x:1000:1000:Student User
One:/home/user1:/bin/bash
- Christine:x:1001:1001::/home/Christine:/bin/bash
- \$
- **\$ cat BashUsers.txt**
- root:x:0:0:root:/root:/bin/bash
- user1:x:1000:1000:Student User
One:/home/user1:/bin/bash
- Christine:x:1001:1001::/home/Christine:/bin/bash
- \$