

# Шелл для мультиварок

или

как перестать бояться и полюбить консоль

# О себе

- Excelsior LLC, 2009-2019

Проект Excelsior JET

- Рантайм-инженер
- QA-lead



- Huawei Technologies Co., Ltd, 2019-...

Excelsior@Huawei, Compilers and Programming Languages Lab



















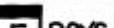
- Всё
- то
- же
- самое



# Зачем нужно знание shell'a и bash'a?

минутка философии

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE  
EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?  
(ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK					
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
HOW MUCH TIME YOU SHAVE OFF	1 SECOND	 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
	5 SECONDS	 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
	30 SECONDS	 4 WEEKS	 3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
	1 MINUTE	 8 WEEKS	 6 DAYS	 1 DAY	4 HOURS	1 HOUR	5 MINUTES
	5 MINUTES	9 MONTHS	 4 WEEKS	 6 DAYS	21 HOURS	5 HOURS	25 MINUTES
	30 MINUTES		6 MONTHS	 5 WEEKS	 5 DAYS	 1 DAY	2 HOURS
	1 HOUR		10 MONTHS	2 MONTHS	 10 DAYS	 2 DAYS	5 HOURS
	6 HOURS				2 MONTHS	 2 WEEKS	 1 DAY
	 1 DAY					 8 WEEKS	 5 DAYS

# План

- Ориентация на местности
  - Оpozнание окружения
  - Работа с файлами
- Процессы
  - Осмотр и взаимодействие
  - Piping
  - Background Jobs
- Bash Advanced
  - Пересказ мануала в свободной форме
- Полезные команды и инструменты
  - Упрощаем себе жизнь

# Ориентация на местности

или

```
user@localhost:~$ what-should-i-do-now?
```

# Прежде чем пойдём вперед. Источники знаний

1. `$ man <command>`
2. `$ man <command>`
3. <https://google.com>
4. <https://stackoverflow.com>
5. `$ man bash`
6. Обратитесь к районному линуксоиду.

# Где я?

**\$ pwd** # показать текущую директорию

/home/username

**\$ ls** # список файлов в директории

Documents hs\_err\_pid1142155.log java\_pid2009.hprof Music Public  
Snapshots  
tmp Videos Desktop Downloads Pictures snap Templates tools

**\$ ls -la**

```
total 6445908
drwxr-xr-x 27 username username 4096 сен  6 15:47 .
drwxr-xr-x  3 root      root      4096 июл  8 22:38 ..
-rw-----  1 username username 33144 сен  7 15:52 .bash_history
-rw-r--r--  1 username username 3987  июл 15 12:58 .bashrc
drwxr-xr-x 18 username username 4096  авг  6 14:57 .config
```

...



# Юзеры и права доступа

- user:group
  - `/home/user` - стандартная домашняя директория
  - У каждого пользователя есть одноименная группа
- Права доступа
  - user - group - other
  - r - чтение
  - w - запись
  - x - исполнение, просмотр каталога
  - Восьмеричная форма: 0777 == rwxrwxrwx, r = 4, w = 2, x = 1
- root == super user == можно всё
  - `/root` - домашняя директория
  - За логин под рутом сисадмины бьют по рукам

# Юзеры и права доступа

```
$ chown user:group file1 file2 # смена владельца
```

```
$ chmod +x myscript.sh # сделать myscript.sh исполняемым
```

```
$ chmod -R 777 /* # режим экстибициониста
```

```
$ chmod g+w /var/log # дать группе право на запись /var/log
```

```
$ sudo ls # исполнить ls из-под root'a
```

```
$ su - # стартовать шелл под root'ом
```

```
$ su other_user # шелл под other_user, сохранив окружение
```

# Файлы

`$ touch ./new_file` # обновить время доступа, создать файл

`$ file /bin/ls` # информация о файле `/bin/ls`

`$ cat /etc/passwd` # вывести содержимое `/etc/passwd`

`$ cat /etc/passwd /etc/crontab` # вывести содержимое файлов

`$ less /etc/bash.bashrc` # удобная читалка (помощь - `h`)

`$ more /etc/bash.bashrc` # неудобная читалка (помощь - `h`)

`$ tail /var/log/dmesg` # вывести несколько последних строк

`$ mv a b` # перемещение `a` в `b`

`$ cp a b` # копирование `a` в `b`

`$ rm file` # удаление `file`

# Файлы. Редакторы.

- **nano** - самый простой и доступный редактор
  - Просто используйте его, где возможно
  - Seriously, вам его хватит для большинства случаев
- **vim** - для пианистов с короткими пальцами
  - Чтобы выйти, последовательно нажмите **:q!**
  - Мощный комбайн на все случаи жизни
  - **vimtutor** - обучение
- **vi** - для пианистов с короткими пальцами, скучающих по 70-ым
  - Порой, единственное, что есть на старых системах
  - **i** - войти в режим редактирования, **esc** - выйти оттуда, **:wq** - выйти из vi и сохранить
- **emacs** - для пианистов с длинными пальцами

# Директории

```
$ cd to/my/directory # перемещение в to/my/directory
```

```
$ cd - # вернуться в предыдущую директорию
```

```
$ pushd to/my/directory # текущую папку в стек и перейти
```

```
$ popd # вернуться в последнюю директорию на стеке
```

```
$ mkdir existing_dir/new_dir # создать new_dir
```

```
$ mkdir -p new_dir/new_dir2 # создать new_dir, а в ней new_dir2
```

# Архивы (потому что никто не знает, как использовать tar)

```
$ zip archive.zip ./foo/* file1 file2
```

```
$ unzip archive.zip -d ./target_directory
```

```
$ rar a archive.rar ./foo/* file1 file2
```

```
$ unrar x archive.rar
```

```
$ tar czf archive.tar.gz ./foo/* file1 file2
```

```
$ tar cjf archive.tar.bz2 ./foo/* file1 file2
```

```
$ tar cJf archive.tar.xz ./foo/* file1 file2
```

```
$ tar xf archive.tar.gz
```

# Ссылки

- Сильные ссылки (или просто ссылки)

- Все файлы - это сильные ссылки
- Файл существует  $\Leftrightarrow$  На файл есть хоть одна сильная ссылка
- Не занимают место, в отличие от копий

```
$ ln /etc/passwd ./my_clear
```

- Символьные ссылки

- Указывают путь
- Могут быть относительными и абсолютными
- По умолчанию, все следуют (follow) символьным ссылкам

```
$ ln -s /usr/bin/chmod ./chmod-link
```

```
$ cp -s /usr/bin/chmod ./chmod-link
```

# Файловая система

- Всё начинается с корня - /
- К корню примонтированы различные файловые системы
- Примонтировать можно что угодно куда угодно (см. linux FUSE)

```
$ mount # информация о примонтированных системах
```

```
# mount -o user=me //myhost/samba_share ./local_dir # примонтировать  
сетевую папку
```

```
# umount ./local_dir # размонтировать
```

```
$ df -h # информация о доступном дисковом пространстве
```



# Интересные места в файловой системе

- `/bin`, `/usr/bin` и всё из `$PATH` - исполняемые файлы
- `/etc` - хранилище конфигурационных файлов
- `/var/log` - логи системы и сервисов
- `/dev` - виртуальные файлы устройств и прочих сущностей
  - `/dev/null`, `/dev/random`, `/dev/zero` ...
  - `/dev/sda*`
- `/proc` - информация о системе и процессах
  - `/proc/cpuinfo`
  - `/proc/<pid>/environ`, `/proc/<pid>/status`, `/proc/<pid>/limits`, ...
  - `$ man proc`

# Процессы

и всё, что вокруг них крутится

# Осмотр процессов и взаимодействие

`$ ps` # процессы, запущенные в данной сессии

`$ ps aux` # подробный список процессов

`$ ps auxf` # древовидное представление

`$ top` # интерактивный диспетчер задач

`$ htop` # прокаченный диспетчер задач

`$ cat /proc/<pid>/...` # основной кладезь информации

`$ kill 12345` # убить процесс с PID'ом 12345

`$ kill -SIGKILL 12345` # послать SIGKILL процессу

`$ killall java` # убить все процессы с именем "java"

# Запуск программ

- По пути или из `$PATH`

```
$ my_tool # поиск исполняемого файла в $PATH
```

```
$ ./my_tool # запуск программы из текущей папки
```

- Переменные окружения

```
$ export JDK_HOME="/home/user/my_jdk_99"
```

```
$ JDK_HOME="/home/user/my_jdk_99" ant build
```

- Код возврата и трюки с ним

```
$ cat none.txt ; echo $? # получить код возврата
```

```
$ cat none.txt || echo "error!" # обработать ошибку
```

```
$ cat /etc/passwd && echo "SUCCESS" # обработать успех
```

# Запуск программ. Стандартные потоки и piping

```
$ echo "some text" >my_new_file.txt # перезаписать файл
$ echo "more text" >>my_new_file.txt # дописать в файл
$ cat <my_new_file.txt # использовать my_new_file.txt в качестве stdin

$ cat none.txt 1>log.txt 2>&1 # stdout и stderr направлены в log.txt
$ cat none.txt 2>&1 1>log.txt # stderr в консоль, stdout в log.txt

$ cat /var/log/* | grep warning | sort | uniq | less # ???
```

# Запуск программ. Foreground vs Background, Jobs

**\$ ping ya.ru &** # запустить **ping** в *background*

**\$ jobs** # список запущенных *job*'ов

**\$ fg** # вывести на передний план последний *job*

**\$ fg %3** # вывести на передний план *job* номер 3

Ctrl+c - убить текущий процесс (SIGINT)

Ctrl+z - убрать на задний фон и усыпить текущий процесс

Ctrl+d - послать EOF процессу

**\$ bg %5** # разбудить и продолжить *job* номер 5 на заднем фоне

**\$ kill %3** # убить *job* под номером 3

**\$ wait 123 %4** # ждать завершения процесса 123 и *job*'а 4

# Bash Advanced

или

что будет, если прочитать мануал

# Главные горячие клавиши

- **Tab** - автодополнение
- **Up / Down** - вверх/вниз по истории команд



# Основные горячие клавиши

- Перемещение курсора
  - `Ctrl+a` / `Ctrl+e` - в начало/конец строки
  - `Alt+b` / `Alt+f` - на слово назад/вперед
  - `Ctrl+x` `Ctrl+x` - переключиться на альтернативный курсор и обратно
- Редактирование
  - `Ctrl+_` - отмена (Undo)
  - `Ctrl+x` `Ctrl+e` - отредактировать текущую командную строку с помощью `$EDITOR`
  - `Ctrl+u` / `Ctrl+k` - удалить (вырезать) от курсора до начала/конца строки
  - `Ctrl+w` / `Ctrl+d` - удалить (вырезать) от курсора до начала/конца слова
  - `Ctrl+y` - вставить последнее вырезанное
  - `Alt+.` - вставить последний аргумент из истории команд
- История
  - `Ctrl+r` - поиск по истории bash'a
  - `Ctrl+g` - выход из поиска

# Работа с путями

- Шаблоны

- \* раскрывается в "что угодно, в том числе пустая строка"

```
$ ls *.e
```

```
$ ls /etc/*/*.cfg
```

- ? раскрывается в "какой-нибудь символ"

```
$ ls /etc/sub?id
```

- [a-z], [12aA], [:ascii:] раскрываются в указанные символы

```
$ ls /etc/[a-z][a-z][a-z][a-z]
```

- Взятие подпутей

```
$ basename /usr/bin/ls # выведет ls
```

```
$ basename -a /usr/bin/ls /etc/passwd # выведет ls и passwd
```

```
$ dirname /usr/bin/ls # выведет /usr/bin
```

```
$ dirname /usr/bin/ls /etc/passwd # выведет /usr/bin и /etc
```

# Переменные

- Создание

- Локальные

```
$ much_var="so value"
```

- Экспортированные

```
$ export PATH=/home/user/script:${PATH}
```

```
$ PATH=/foo:${PATH} some_var="value" ./my_script.sh
```

- Использование

- `${much_var}`, `$much_var` -> `so value`
- `${no_var:-"default $much_var"}` -> `default so value` (`no_var` не задан)
- `${no_var:="default $much_var"}` -> `default so value` (в `no_var` записано значение)
- `${much_var/value/good}` -> `so good`

- Удаление

```
$ unset much_var
```

# Кавычки это сложно

- Два вида кавычек
  - `"` - позволяют раскрытие переменных и выражений
  - `'` - буквальное значение
- Escape-последовательности
  - `\$, \', \", \` (тут пробел)...

- Кавычки не сохраняются!

- Примеры

```
$ echo "$PATH" # выведется содержимое $PATH
```

```
$ echo '$PATH' ; echo "\$PATH" # в обоих случаях выведется $PATH
```

```
$ touch 1 2 ; touch '1 2' # первая - два файла 1 и 2, вторая - файл '1 2'
```

```
$ touch 1\ 2 # аналог вызова touch '1 2'
```

```
$ f="1 2" ; touch $f # аналог вызова touch 1 2
```

```
$ f="1 2" ; touch "$f" # аналог вызова touch "1 2"
```

# Особые переменные

- `$1`, `$2` ... - параметры скрипта/функции
- `$*` vs `$@` - список всех аргументов
  - `"$*" -> "$1 $2 $3 ..."`
  - `"$@" -> "$1" "$2" "$3" ...`

`$ shift 3` # переименовать `$4` и далее в `$1` итд

- `$0` - имя скрипта
- `$#` - количество параметров
- `$?` - код возврата последней команды
- `!` - pid последней команды, запущенной на заднем фоне

# Особые выражения

- Исполнение под-команд

```
$ echo $(ls -la) # выведет результат команды ls -la
```

```
$ echo `ls -la` # синоним (используются обратные кавычки)
```

- Арифметика

```
$ echo $((100 + (31 * 0xabc) - 0777)) # выведет 84777
```

- Булевы выражения

```
$ [[ 3 > 2 ]] && echo "true" # проверка арифметики
```

```
$ [[ "foo" == "bar" ]] || echo "false" # проверка строк
```

```
$ [[ -e /etc/passwd ]] && echo "true" # проверка наличия файла
```

```
$ [[ ! ( -z $non_existant) ]] || echo "false" # проверка, что  
строка не пуста
```

# Контролирующие операторы. Условный оператор if

```
$ if [[ 3 > 2 ]]; then echo 1; elif [[ 2 > 3 ]] ; then echo 2; else echo 3; fi
```

```
$ if cat none.txt ; then # можно использовать любую команду  
> echo 1  
> ls -la  
> else  
> echo 2  
> fi # многострочная команда
```

```
$if (cat none.txt) ... # скобки для тех, кто любит порядок
```

# Контролирующие операторы. Циклы

```
$ for f in `ls` ; do echo $f ; done
```

```
$ for (( i=10 ; i ; i-- )) ; do echo $i ; done
```

```
$ for (( i=0 ; i < 10 ; i++ )) ; do echo $i ; done
```

```
$ while [[ 3 > 2 ]] ; do echo INFINITY ; done
```

```
$ until [[ 3 > 2 ]] ; do echo NEVER ; done
```



# Контролирующие операторы. Case

```
$ case "$mychoice" in
> "apple") # допустимы обычные строки
> echo apple
> ;; # ￣\_ (ツ) _/￣
> "?rang*") # шаблоны
> echo $mychoice
> ;;
> *) # да и кавычки необязательны
> echo something else
> ;;
> esac # (ノ◻ノ) ノ ￣￣
```

## Функции. Проще некуда.

```
$ function test_my_knowledge() { # скобки скорее для красоты
> for a in "$@"; do
> echo $a
> done
> return 42 # код возврата
> }
```

```
$ test_my_knowledge a b c ; echo $?
```

# Скрипты

- Просто последовательность команд
- Шелл - указатель исполняемой команды

```
#!/bin/bash
```

```
#!/bin/sh
```

```
#!/bin/bash --help
```

- Не забудьте **+x**!
- Исполнение

```
$ ./my_script.sh # исполнение в отдельном процессе
```

```
$ source ./my_script.sh # исполнение в текущем шелле
```

```
$ . ./mys_script.sh # то же самое, что и source
```

# So much more

- Еще много интересных вещей
  - Алиасы: `$ alias ll=ls -la`
  - Массивы: `$ array_var[23]=val, $ echo ${array_var[*]}`
  - Here Documents: `$ cat <<FINISH`
  - Специальные переменные окружения: `$RANDOM`
  - Ссылки на историю команд: `$ echo !!, $ !9, $ !grep`
  - Вариации: `$ cp archive.zip{,.bak}`
- Что почитать на тему
  - `$ man bash`
  - [https://www.opennet.ru/docs/RUS/bash\\_scripting\\_guide/](https://www.opennet.ru/docs/RUS/bash_scripting_guide/)

Полезные команды и инструменты  
если ещё остались силы и время

# Полезные команды на все случаи жизни

- Поиск в файлах с помощью регулярных выражений

```
$ grep "error" log.txt
```

```
$ grep -E -i 'error|warning' log.txt
```

- Поиск файлов

```
$ find /etc -name 'pass*'
```

```
$ find /etc -name '*pass*' -exec echo found '{} ' \;
```

- Выяснить, кто сожрал диск

```
$ du -chd1 .
```

- Отредактировать поток текста

```
$ cat log.txt | sed 's/error/success/g'
```



# Полезные команды на все случаи жизни

- Как опознать линукс

```
$ uname -a
```

```
$ cat /etc/*release
```

- Поиск пакетов в разных менеджерах

```
$ apt search gcc # Ubuntu, Debian...
```

```
$ yum search gcc # RedHat, CentOS, Fedora...
```

```
$ pacman -Ss gcc # ArchLinux
```

- Информация о сетевых интерфейсах

```
$ ip addr
```

```
$ ifconfig -a # считается устаревшим
```

# Полезные команды на все случаи жизни

- Вытащить что-то из интернета

```
$ wget http://google.com # скачает файл/страницу
```

```
$ curl http://google.com # выведет ответ от сервера в stdout
```

```
$ scp user@192.168.0.45:/etc/passwd . # копирование по SSH
```

- Удобная работа с output'ом

```
$ ./my_service 2>&1 | tee out.log # дублировать вывод в out.log и в консоль
```

- И вообще, все команды не перечислишь

- <https://www.lancaster.ac.uk/staff/tipper/writing/luui/html/unix-intro012.html>



# Файловый менеджер mc (Midnight Commander)

```
Left      File      Command      Options      Right
<-- ~ -->                                     <-- /etc -->
.n                Name                Size  Modify time
UP--DIR Sep 11 06:47
/.cache           4096 Sep 12 07:09
/.config          4096 Sep 12 07:09
/.local           4096 Sep 11 11:28
/avatar           4096 Sep 11 08:08
/fo               4096 Sep 11 08:35
.Xauthority       50 Sep 11 11:35
.bash_history     3778 Sep 12 07:09
.bash_logout     220 Feb 25 2020
.bashrc           3771 Feb 25 2020
.lessshst        72 Sep 12 05:36
.profile          807 Feb 25 2020
.sudo_as_admin_successful 0 Sep 11 06:54
.viminfo         772 Sep 12 06:08
60               2 Sep 12 05:39
63               2 Sep 12 05:39
fo.tar.gz        179 Sep 11 08:35
foo.log          610 Sep 11 11:10
foo.txt          0 Sep 12 05:39
index.html      16784 Sep 12 06:58
l               0 Sep 11 10:36
new.txt         48 Sep 11 10:42
rrr             0 Sep 11 10:37
*test.sh        26 Sep 12 06:08

UP--DIR                                     UP--DIR
126/20G (63%)                             126/20G (63%)

Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.
xappymah@demo:/etc$
1Help      2Menu      3View      4Edit      5Copy      6RenMov     7Mkdir     8Delete     9PullDn    10Quit
```

# Мультиплексер терминала Tmux

```
xappymah@demo:~$ cowsay "Tmux xopow!"
```

< Тмух хорош! >

```

      ^ ^
      (oo)\_____
      ( _)\         )\/\
      ||-----w||
xappymah@demo:~$

```

```
xappymah@demo:~$
```

Left		File	Command	Options	Right	
~		.[^]			~	
.n	Name	Size	Modify	time	.n	Name
/..		UP--DIR	Sep 12 06:47		/..	
/.cache		4096	Sep 12 07:09		/.cache	
/.config		4096	Sep 12 07:09		/.config	
/.local		4096	Sep 11 11:28		/.local	
/avatar		4096	Sep 11 08:08		/avatar	
/fo		4096	Sep 11 08:35		/fo	
.Xauthority		50	Sep 11 11:35		.Xauthority	
.bash_history		3778	Sep 12 07:09		.bash_history	
.bash_logout		220	Feb 25 2020		.bash_logout	
.bashrc		3771	Feb 25 2020		.bashrc	
UP--DIR		12G/20G (63%)			UP--DIR	
					12G/20G (63%)	

Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.

```
xappymah@demo:~$
```

1Help 2Menu 3View 4Edit 5Copy 6Re~ov 7Mkdir 8Delete 9PullDn10Quit

```

1 [          0.6%] Tasks: 47, 35 thr; 1 running
2 [          0.0%] Load average: 0.03 0.07 0.02
Mem [|||||||] 203M/3.84G Uptime: 1 day, 00:30:10
Swp [          0K/3.84G]

```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1	root	20	0	100M	12828	8324	S	0.0	0.3	0:02.64	/sbin/init
8545	root	20	0	275M	16276	14356	S	0.0	0.4	0:00.01	/usr/lib/
8549	root	20	0	275M	16276	14356	S	0.0	0.4	0:00.00	/usr/l
8546	root	20	0	275M	16276	14356	S	0.0	0.4	0:00.00	/usr/l
8391	xappymah	20	0	7520	4012	3176	S	0.0	0.1	0:00.10	tmux
8412	xappymah	20	0	8284	5276	3556	S	0.0	0.1	0:00.01	-bash
8422	xappymah	20	0	7872	3964	3384	R	0.0	0.1	0:00.22	htop
8402	xappymah	20	0	8284	5224	3504	S	0.0	0.1	0:00.01	-bash
8423	xappymah	20	0	19308	8860	7240	S	0.0	0.2	0:00.00	mc
8427	xappymah	20	0	8284	5008	3384	S	0.0	0.1	0:00.00	l
8392	xappymah	20	0	8284	5172	3452	S	0.0	0.1	0:00.01	-bash

F1Help F2Setup F3SearchF4FilterF5SortedF6CollapF7Nice -F8Nice +F9Kill F10Qu

```
[0] 0: bash*
```

# Спасибо за внимание

```
# shutdown -h now
```