



Команды оболочки

Unix-подобных операционных систем

Формат команд:

команда -ключ1 -ключ2 -ключN параметр1 параметр2 параметрN

Элементы команды разделяются (одним или несколькими) пробелами; если элемент включает пробелы, его заключают в 'одинарные кавычки', а если требуется подстановка по шаблону, то в "двойные кавычки".

Специальные обозначения в командах:

Описание	Обозначение	Пример	Пояснения
Разделитель элементов команды	пробел	cp 1.txt /tmp	Скопировать 1.txt в /tmp
«Ключ» / «флаг» = указание / режим работы команды	<i>-флаг или --режим</i>	ls -a	-a = «All» - все файлы
Разделитель каталогов в пути	<i>/каталог/подкаталог</i>	/home/pi	
Корень файловой системы	<i>/</i>	cd /	Рабочим каталогом станет /
Скрытый файл или каталог (папка)	<i>.имя</i>	cd .config	Видны по ls -a
Текущий каталог	<i>.</i>	ls .	То же, что просто ls
Родительский каталог (каталог на 1 уровень выше)	<i>..</i>	cd ..	Рабочим станет надкаталог
Домашний каталог (папка) пользователя	<i>~</i>	cd ~	Рабочим станет домашний
Шаблон для одного любого символа в имени	<i>?</i>	cat ?.txt	1.txt a.txt, но не abc.txt
Шаблон для нескольких любых символов в имени	<i>*</i>	cat *.txt	1.txt a.txt abc.txt
Отмена специального значения символа после \	<i>\</i>	mkdir Moi\ файлы	То же, что mkdir 'Мои файлы'
Переменная	<i>\$имя</i>	\$PATH	Пути для поиска исполняемых файлов
Параметр как единая строка с подстановкой по шаблону	<i>"строка"</i>	echo "\$PATH"	/home/pi/bin:/usr/local/bin:/usr/bin:/bin
Параметр как единая строка с подстановкой по шаблону	<i>'строка'</i>	echo '\$PATH'	\$PATH
Подставить результат выполнения команды	<i>`команда`</i>	echo `date` DIR=`date`	Пн ноя 12 05:12:25 +05 2018 В переменной DIR будет дата.

Перенаправить вывод команды в файл (перезаписать в файл)	<i>команда > файл</i>	cat ?.txt > all.txt	all.txt=1.txt+2.txt+a.txt
Перенаправить вывод команды в файл (дописать в файл)	<i>команда >> файл</i>	cat ???.txt >> all.txt	all.txt=all.txt+10.txt+20.txt
Направить данные из файла на вход команды (читать из файла)	<i>команда < файл</i>	sort < 1.txt	То же, что cat 1.txt sort
Передать вывод одной команды на вход другой	<i>команда1 команда2</i>	cat *.txt sort	Отсортировать то, что передала cat
Запустить команду на выполнение в фоновом режиме	<i>команда &</i>	server &	Запустится как служба
Выполнить команды одну за другой	<i>команда1 ; команда2</i>	mkdir x ; mkdir z	Создаст x/, потом z/
Выполнить 2-ю команду при успешном завершении 1-й	<i>команда1 && команда2</i>	mkdir x && echo OK	OK, если каталога x/ не было
Выполнить 2-ю команду при аварийном завершении 1-й	<i>команда1 команда2</i>	mkdir x echo FAIL	FAIL, если каталог x/ уже был

Часто используемые команды Unix / Linux

Описание	Команда	Пример	Пояснения
Файловая система: каталоги и файлы			
Имя рабочего (текущего) каталога (Print Work Directory)	pwd	pwd	Выведется полный путь
Список файлов в текущем каталоге (LiSt)	ls	ls -a	Все, в т. ч. скрытые
		ls -l	Полная информация
		ls -ld /home/pi	Полная инфо о каталоге
		ls *.sh	Только об sh-файлах
Создать (пустой) файл	touch <i>файл</i>	touch README.txt	Файл нулевой длины
		touch ReadMe.txt	Заглавные и строчные буквы в именах отличаются!!!
		touch readme.txt	
Создать подкаталог (MaKe DIRectory) в рабочем каталоге	mkdir <i>каталог</i>	mkdir bin	Подкаталог ./bin
Создать каталог с указанием полного пути	mkdir <i>/путь/до/каталога</i>	mkdir /home/pi/bin	
Перейти в каталог (Change Directory)	cd <i>каталог</i>	cd bin	Текущим станет ./bin
Перейти на один каталог выше (в дереве файловой системы)		cd ..	Текущим станет родительский
Перейти в корень файловой системы		cd /	Текущим станет /
Перейти в домашний каталог пользователя		cd ~	Для pi будет /home/pi
Перейти в подкаталог в домашнем каталоге		cd ~/bin	Текущим станет bin
Перейти в каталог по абсолютному имени (считая от корня)		cd /home/pi/bin	Текущим станет bin
Скопировать содержимое файла в другой (CoPy)	cp <i>файл1 файл2</i>	cp README.txt README.old	./README.old
Скопировать файл (с тем же именем) в каталог	cp <i>файл каталог</i>	cp README.txt ~/doc	~/doc/README.txt
Скопировать каталог (с содержимым) в другой каталог	cp -r <i>каталог1 каталог2</i>	cp -r ~/doc /tmp	/tmp/doc
Переместить файл или каталог (MoVe)	mv <i>имя куда</i>	mv README.txt /tmp	cp README.txt /tmp rm README.txt
Переименовать файл или каталог	mv <i>имя1 имя2</i>	mv README.rdoc readme.txt	
Удалить файл (ReMove)	rm <i>файл</i>	rm README.rdoc	
Удалить пустой каталог (ReMove DIRectory)	rmdir <i>каталог</i>	rmdir ./tmp	Не пустой не удалится
Принудительно (Force) удалить непустой каталог со всеми	rm -rf <i>каталог</i>	rm -rf ~/tmp	Осторожно: удалится всё

подкаталогами (Recursive)			поддерево каталогов!!!
Информация о командах, файлах, процессах, времени, пространстве			
Выдать справку (MANual) по команде	man команда	man sort	Выход — клавиша q
Показать полный путь до исполняемого файла	which файл	which bash	/bin/bash
Показать содержимое файлов (на экране)	cat файл1 файл2 ...	cat README.txt	Все строки файла
Объединить содержимое файлов (и записать в файл)	cat файл1 файл2 > файл	cat 1.txt 2.txt 3.txt > all.txt	Всё из ?.txt → all.txt
Вывести строки, содержащие выражение по шаблону	grep шаблон файл1 файл2	grep disk /var/log/boot.log	Только строки с «disk»
Показать список процессов (ProceSs)	ps	ps ax	
Показывать вывод постранично (поэкранно)	команда more	ps ax more	Пробел = следующая страница
Вывести текущую дату и время	date	date	Пт ноя 12 12:25:55 +05 2018
Вывести текущую дату и/или время в указанном формате	date +"%Y-%m-%d %H:%M:%S"	date +"%d.%m.%y %H:%M"	
Вывести сообщение (на экран)	echo текст	echo "Сегодня: `date -l`"	Сегодня: 2018-11-12
Копировать вход в каждый файл и в стандартный вывод	tee файл1 файл2 ...	echo OK tee log.txt	ОК на экран и в файл
Показать информацию о файле (размер, время изменения, доступа, ...)	stat имя	stat /home/pi	
Список смонтированных файловых систем	mount	mount	
Сведения о свободном пространстве на карте памяти, дисках	df	df -h	-h = «по-человечески»
Сведения о свободной оперативной памяти	free	free -h	
Просмотр списка (LiSt) подключенных устройств USB	lsusb	lsusb -tv	-v = подробно
Информация об устройствах USB	usb-devices	usb-devices	
Показать версию ядра ОС и системную архитектуру	uname	uname -a	-a всю информацию
Очистить экран терминала	clear	clear	
Права доступа и владения на каталоги и файлы			
Сведения о пользователе (UID) и его группах (GID)	id пользователь	id	UID и GID для pi
Изменить права доступа файла или каталога (CHange MODe)	chmod кому±права имя		
Добавить (+) для всех (All) право выполнения (eXecute)		chmod a+x *.sh	sh-файлы - исполняемые
Добавить (+) членам группы (Group) право записи (Write)		chmod g+w /home/iot/projects	

Убрать (-) у не-членов группы (Other) право чтения (Read)		chmod o-r /home/iot/projects	
Назначить (CHange) владельца (OWNer) файлов	chown <i>владелец:группа имя1</i>		
Назначить pi владельцем каталога iot/ с подкаталогами (-R)		chown -R pi iot	
Выполнить команду от имени супер-пользователя	sudo <i>команда</i>	sudo reboot	Перезагрузить ОС
Работа с сетью			
Просмотр (и изменение) сетевых настроек	ifconfig <i>интерфейс</i>	ifconfig eth0	Проводной адаптер
Просмотр всех сетевых настроек		ifconfig	Все сетевые интерфейсы
Сетевое имя этого компьютера (узла сети)	hostname	hostname	
IP-адрес (-I) этого компьютера		hostname -I	
Проверка доступности узла сети	ping <i>узел</i>	ping -c 5 yandex.ru	Пингануть 5 раз
Подключение к терминалу на удалённом компьютере	ssh <i>пользователь@узел</i>	ssh pi@raspberry	
Управление компьютером			
Перезагрузить компьютера немедленно	reboot	sudo reboot	
Остановить компьютера немедленно	halt	sudo halt	
Выключить компьютера немедленно	poweroff	sudo poweroff	
Перезапустить, остановить, выключить компьютер (в заданное время)	shutdown <i>действие время</i>	sudo shutdown -r now sudo shutdown -H +10 sudo shutdown -P "23:59"	Перезагрузить сейчас. Остановить через 10 мин. Отключить в 23:59.
Управление программными пакетами, загрузка программ			
Найти название программного пакета	sudo apt-cache search <i>пакет</i>	sudo apt-cache search mc	
Обновить базу данных пакетов программ	sudo apt-get update	sudo apt-get update	
Обновить все установленные пакеты	sudo apt-get upgrade	sudo apt-get upgrade	Y =согласиться
Установка пакета программ и зависимых пакетов	sudo apt-get install <i>пакет</i>	sudo apt-get ruby-dev	*-dev = для разработчика
Удалить программный пакет	sudo apt-get remove <i>пакет</i>	sudo apt-get remove mc	Чтобы освободить место
Обновить дистрибутив ОС Raspbian до последней версии	sudo apt-get dist-upgrade	sudo apt-get dist-upgrade	
Установить пакет для языка программирования Ruby	sudo gem install <i>пакет</i>	sudo gem install ruby-gpio	
Скачать программы (обычно, архив) с сайта	wget <i>сайт/путь/файл</i>	wget http://site/path/file.tar.gz	В текущий каталог
Загрузить каталог проекта из системы хранения версий	git clone <i>https://адрес.git</i>	git clone https://github.com/x/z.git	В текущий каталог

Информация о аппаратных и программных средствах			
Сведения о центральном процессоре (CPU)	cat /proc/cpuinfo	cat /proc/cpuinfo	
Температура SoC по встроенному датчику температуры	vcgencmd measure_temp	vcgencmd measure_temp	
Сведения о распределении оперативной памяти (ОЗУ) между CPU и GPU (графическим процессором)	vcgencmd get_mem <i>процессор</i>	vcgencmd get_mem arm vcgencmd get_mem gpu	
Сведения о свободной оперативной памяти	free	free -h	
Сведения о портах GPIO	gpio <i>команда</i>	gpio readall	Схема контактов GPIO

<<https://www.maketecheasier.com/finding-raspberry-pi-system-information/>>

<https://elinux.org/RPI_vcgencmd_usage>

<<https://www.pantz.org/software/cron/croninfo.html>>

Управляющие конструкции в командных файлах `sh` / `bash`

Правила записи командных файлов (скриптов):

- Заглавные и строчные буквы отличаются.
- Команды записываются по одной на каждой строке.
- Если нужно записать несколько команд на одной строке, они разделяются символом «точка с запятой» (;).
- Каждая управляющая конструкция имеет начальное и завершающее *ключевое слово* (if — fi, case — esac, do — done).
- Не следует использовать имена переменных, совпадающих с ключевыми словами *управляющих конструкций*.

Описание	Формат	Пример
Комментарий однострочный	<code># текст</code>	<code># Это комментарий</code>
Имя программы-интерпретатора для выполнения (shebang)	<code>#!/путь/к/интерпретатору</code>	<code>#!/bin/bash</code>
Переменная: присваивание значения	<code>имя=значение</code>	<code>PIN=21</code> <code>result=`gpio -g read \$PIN`</code>
Переменная: использование значения	<code>\$имя</code>	<code>echo \$result</code>
Проверка условия	<code>[условие]</code>	<code>if [! -s "\$file"]; then</code> <code>echo "Файл \$file - нулевой длины!"</code> <code>fi</code>
Ветвление по условиям	<code>if условие1</code> <code>then</code> <code>команда1</code> <code>elif условие2</code> <code>then</code> <code>команда2</code> <code>else</code> <code>команда3</code> <code>fi</code>	<code>if [! -e "\$file"]; then</code> <code>echo "Файл \$file не найден."</code> <code>fi</code> <code>if [-d "\$name"]; then</code> <code>echo "\$name - это каталог"</code> <code>else</code> <code>echo "\$name - это файл"</code> <code>fi</code>
Операции сравнения целых чисел	<code>-eq</code> # равно (Equal) <code>-ne</code> # не равно (Not Equal) <code>-gt</code> # больше (Greater Than) <code>-ge</code> # больше или равно (Greater or Equal) <code>-lt</code> # меньше (Less Than) <code>-le</code> # меньше или равно (Less or Equal)	<code>if [\$temp -ge 78000]; then</code> <code>echo Перегрев процессора!!!</code> <code>fi</code>
Операции сравнения строк	<code>==</code> # равно <code>=</code> # равно <code>!=</code> # не равно	<code>if ["\$a" == "\$b"]; then</code> <code>echo \$a равно \$b</code> <code>fi</code>

	<pre> > # больше >= # больше или равно < # меньше <= # меньше или равно -Z # строка "пустая", имеет нулевую длинну -n # строка не пустая </pre>	<pre> if [-n \$string1]; then echo "Строка не пустая." else echo "Строка пустая." fi </pre>
Выбор команды по условию	<pre> case \$переменная in значение1) команда1 ;; значение2) команда2 ;; esac </pre>	<pre> case \$fruit in apple) echo 'Люблю яблоки!' ;; banana) echo 'И бананы тоже.' ;; orange tangerine) echo 'А цитрусовые — нет!' ;; *) echo 'А это что за фрукт?' esac </pre>
Повторение: цикл перебора элементов списка	<pre> for элемент in [список] do команды done </pre>	<pre> for language in RU EN FR ES JP CN; do echo \$language done for file in *; do echo "Размер `stat -c '%s' \$file`" done </pre>
Повторение: цикл с условием продолжения (повторять, пока условие истинно)	<pre> while условие do команды done </pre>	<pre> while true; do # бесконечный цикл done while ["\$var1" != "end"]; do echo "Введите значение:" echo "(end – выход)" read var1 echo "значение = \$var1" echo done </pre>
Повторение: цикл с условием окончания (повторять, пока условие ложно)	<pre> until условие do команды done </pre>	<pre> until ["\$var1" = "end"]; do echo "Введите значение:" echo "(end – выход)" read var1 done </pre>

		echo "значение = \$var1" done

<https://www.opennet.ru/docs/RUS/bash_scripting_guide/>