Bash ????? ??????????

Консоль работает с текстом, поэтому центральная задача - обработка текста.

echo	Просто выводит строку
printf	Выводит строку с модификаторами printf "%s %d" "ff" 22

Поиск приложения: whereis name

~ - домашняя директория, ~sergey домашняя директория пользователя sergey

cd без аргументов в домашнюю, cd - в предыдущую

Маски для ls

?	один символ
*	много символов
[set]	Набор символов, в данном случае s,e,t [a-c] a,b,c [!0-9] где нет цифр
{}	перечисляются наборы текста, например echo f{io,am}

Утилиты фильтрации текста

cat	перенаправление ввода на вывод -b - нумеровать только непустые строки; -E - показывать символ $ в конце каждой строки; -n - нумеровать все строки; -s - удалять пустые повторяющиеся строки cat file1 file2 > file3

grep	поиск строк во вводе Может анализировать один файл или директорию. -e несколько регулярных выражений, например -e ... -e ... -n номер строки, где найдено совпадение -i, --ignore-case - не учитывать регистр символов; -v, --invert-match - вывести только те строки, в которых шаблон поиска не найден; -w, --word-regexp - искать шаблон как слово, отделенное пробелами или другими знаками препинания; -x, --line-regexp - искать шаблон как целую строку, от начала и до символа перевода строки; -c - вывести количество найденных строк; -L, --files-without-match - выводить только имена файлов, будут выведены все файлы в которых выполняется поиск; -l, --files-with-match - аналогично предыдущему, но будут выведены только файлы, в которых есть хотя бы одно вхождение; -m, --max-count - остановить поиск после того как будет найдено указанное количество строк; -o, --only-matching - отображать только совпавшую часть, вместо отображения всей строки; -h, --no-filename - не выводить имя файла; -s, --no-messages - не выводить ошибки чтения файлов; -A, --after-content - показать вхождение и n строк после него; Те -A2 строка + 2 строки после -B, --before-content - показать вхождение и n строк перед ним; -C - показать n строк до и после вхождения; -a, --text - обрабатывать двоичные файлы как текст; --exclude - пропустить файлы имена которых соответствуют регулярному выражению; --exclude-dir - пропустить все файлы в указанной директории; -I - пропускать двоичные файлы; --include - искать только в файлах, имена которых соответствуют регулярному выражению; -r - рекурсивный поиск по всем подпапкам; -R - рекурсивный поиск включая ссылки; `grep -r "поисковой_запрос" /путь/к/директории/`

sort	сортировка строк -b - не учитывать пробелы -d - использовать для сортировки только буквы и цифры -i - сортировать только по ASCII символах -n - сортировка строк linux по числовому значению -r - сортировать в обратном порядке -o - вывести результат в файл -u - игнорировать повторяющиеся строки -m - объединение ранее отсортированных файлов -k - указать столбец по которому нужно сортировать строки, если не задано, сортировка выполняется по всей строке. ls -l | sort -k9 -f - использовать в качестве разделителя полей ваш символ вместо пробела.
uniq f1 > f2	Удаляет повторяющиеся строки
cut	извлечение символов из ввода -b 2-6, 8 отобразит символы от 2 по 6 из строки и 8 -b 5- символы с 5 до конца -c нечто похожее на -b -d ':' -f 1,3 считать разделителем : и оставить первый и третий столбцы
sed	операции изменения данных перед выводом Позволяет вырезать строки, поиск/замена по регулярке, вывод. Крутая штука.
tr	транслирование символов на вводе в другие символы -d набор - удаление символов tr -d '0-9' -s заменяет последовательность повторяющихся символов в SET1 на один такой символ; tr -s ' ' заменит много пробелов на один
head	-n кол-во Выводит первые кол-во строк

awk	Считывает построчно данные, выполняет действия и выводит в stdout. Включает целый язык для обработки текста. awk опции 'условие {действие} условие {действие}' Опции: -F, --field-separator - разделитель полей, используется для разбиения текста на колонки; -f, --file - прочитать данные не из стандартного вывода, а из файла; -v, --assign - присвоить значение переменной, например foo=bar; -b, --characters-as-bytes - считать все символы однобайтовыми; -d, --dump-variables - вывести значения всех переменных awk по умолчанию; -D, --debug - режим отладки, позволяет вводить команды интерактивно с клавиатуры; -e, --source - выполнить указанный код на языке awk; -o, --pretty-print - вывести результат работы программы в файл; Действия: print(строка) - вывод чего либо в стандартный поток вывода; printf(строка) - форматированный вывод в стандартный поток вывода; system(команда) - выполняет команду в системе; length(строка) - возвращает длину строки; substr(строка, старт, количество) - обрезает строку и возвращает результат; tolower(строка) - переводит строку в нижний регистр; toupper(строка) - переводить строку в верхний регистр. Переменные: $NF - последняя строка $(NF-1) - предпоследняя NR - количество строк с начала awk -F":" '{print $4}' Возьмет строку и по разделителю : выведет 4 элемент Поддерживает регулярные выражения echo -e 'one 1\n two 2' | awk '{sum+=$2} END {print sum}' сумма элементов. awk 'NR < 10' log.txt
sed	Тоже очень крутая штука, этим двум командам посвящена целая книга.
diff file1 file2	Различие в строках
ndiff	Различие в результатах сканирования nmap `${BIN_PATH}ndiff $BASE_RESULTS $NEW_RESULTS > $NDIFF_RESULTS`

jq	Утилита для анализа JSON. jq -r '.first' возвращает значение ключа first jq '.[0].plugins.HTTPServer.string[0] определяет следующий элемент: `[ { "plugins": { "HTTPServer": { "string": [ "Werkzeug/3.0.1 Python/3.12.3" ] } } } ]`
tail -F filename	Отображения изменения в файле
wget	Загрузка файлов -q тихий режим -r рекурсивное скачивание -np скачивание файлов из данной директории или ниже -R "index.html" исключить загрузку файлов вида index.html -P foldername директория сохранения

curl	Must have.
	-s тихий режим
	-X GET|POST|PUT|DELETE → явный выбор метода
	-d "param=value" → данные формы (POST)
	`curl -X POST -H "Content-Type: application/json" \ -d '{"name":"Alice"}' https://api.example.com/users`
	
	-H "Header: value" → добавить заголовок
	`curl -H "Authorization: Bearer TOKEN" https://api.example.com`
	
	-b "name=value" → передать cookie
	-c cookies.txt → сохранить cookie в файл
	-b cookies.txt → использовать cookie из файла
	-o file → сохранить в файл
	-u user:pass → basic auth
	-k → игнорировать ошибки сертификатов (небезопасно!)
	--cacert file.crt → указать свой сертификат CA
	-w "%{http_code}\n" → вывести только HTTP-код

find / -name "file.txt"	Поиск файла

Редиректы

Вывод в файл:

Оператор	Значение	Пример
`>`	Перенаправить вывод (перезаписать файл)	`echo "hi" > file.txt`
`>>`	Перенаправить вывод (добавить в файл)	`echo "hi" >> file.txt`

Вывод из файла:

Оператор	Значение	Пример
`<`	Чтение из файла	`wc -l < file.txt`
`<<`	**Here document** — передаёт блок текста как stdin	cat << EOFстрока1строка2EOF
`<<<`	**Here string** — передаёт одну строку (или переменную)	`grep foo <<< "foo bar baz"`

Фоновые задачи

Фоновые задачи не получают данные от клавиатуры

Команда	Что делает
`command &`	Запускает процесс в фоне
`jobs`	Показывает список фоновых задач (с job ID: `%1`, `%2` …)
`fg %N`	Переводит задачу номер N в передний план
`bg %N`	Возобновляет приостановленную задачу в фоне
`Ctrl+Z`	Приостанавливает текущий процесс (отправляет в «stopped»)
`kill %N`	Завершает задачу по её job ID
`kill PID`	Завершает задачу по PID
`kill -9 PID`	Жёсткое завершение (если обычный `kill` не помог)
`disown %N`	«Отвязывает» задачу от текущего терминала (будет жить после выхода)
`ps aux	grep name`	
`pkill -f pattern`	Завершение процессов по шаблону

Задачи завершаются при закрытии сессии. Через команду nohup можно сделать задачу, работающую после закрытия сессии.

```
nohup ./myscript.sh &
```

История ввода команд

Хранится где-то в сессии, сохраняется в .bash_history при закрытии сессии. Для просмотра введенных команд в пределах сессии используется утилита fc

-l	Последние 15 команд 4 выведет четвертую команду 4 7 команды с 4 по 7 строка - выведет с первого вхождения строки (из 15) и далее
-n	без вывода номеров
-e nano	без дальнейших аргументов - редактирование последней команды и затем исполнение Чтобы не вводить редактор, можно в .bash_profile добавить переменную FCEDIT=nano

!! исполнение предыдущей команды

Специальные файлы

.bash_profile, .bash_logout, .bashrc Если отсутствуют - используются файл /etc/profile и файлы из /etc/profile.d

.bash_profile - сейчас не используется.

В .profile переменные окружения, .bashrc - скрипты. .bash_login при входе. Но идея осталась.

Последовательность поиска при авторизации, исполняется первый найденный

- `~/.bash_profile`
- `~/.bash_login`
- `~/.profile`

Исполняет команды при каждом входе в систему.

Итого:

~/.bashrc

Используется в интерактивных (обычных) shell-ах. Сюда кладём то, что нужно только в терминале:

- алиасы (alias ll='ls -lh')
- функции (mkcd() { mkdir -p "$1" && cd "$1"; })
- настройки PS1 (prompt, цвета)
- shopt и настройки истории (HISTSIZE, HISTCONTROL)
- подключение fzf, автодополнений и прочих «тюнингов»

~/.profile

Используется при login-shell (и не только Bash, а sh/dash/zsh тоже могут его читать). Сюда кладём:

- переменные окружения (PATH, EDITOR, LANG, JAVA_HOME)
- настройки, которые должны работать и в GUI-программах (например, экспорт GTK_THEME или XMODIFIERS)
- запуск программ при логине (например, ssh-agent или gnome-keyring-daemon)

~/.bash_profile или ~/.bash_login

Сегодня почти не нужны. Если они есть, то Bash не будет читать ~/.profile.

source .profile - перечитать файл

Alias, options и переменные окружения

Алиасы для команд. Создание алиасов: alias cdvoy='cd sipp/demo/animation/voyager'

Пробел перед закрывающей ' означает ожидание ввода пользователя.

unalias name - удаление

set +o optionname установить опцию, -o убрать. Фиксированный набор. Не впечатлило.

Переменные

Синтаксис: varname='something', использование $varname удаление unset varname

Формат timestamp:

%a, %A	Аббревиатура дня недели, Полное имя дня недели
%b, %B, %m	Аббревиатура имени месяца, Полное имя месяца, Номер месяца в числовом формате
%c	Дата и время локали
%C	Последние 2 цифры года
%H, %I, %p	Час в 24-часовом формате, Час в 12-часовом формате, эквивалент am/pm
%d, %e	Цифра дня, цифра дня с пробелом в случае одной цифры
%D	Дата в американском формате (%m/%d/%y)
%j	День года 001-366
%M	Минута в десятичном представлении
%n, %t	Новая строка, табуляция
%R	Время в 24-часовом формате
%S	Секунда в десятичном формате
%T	Время (час:минута:секунда)
%u	Номер дня недели в десятичном формате
%U	Номер недели в году
%Y	Год

Строка приглашения

PS1 - основное приглашение

PS2 - при незавершенной строке

В переменной PS1

Код	Что показывает
`\u`	Имя текущего пользователя
`\h`	Имя хоста (до первой точки)
`\H`	Полное имя хоста
`\w`	Текущая директория (полный путь, с `~` для home)
`\W`	Текущая директория (только имя последней папки)
`\!`	Номер текущей команды в истории
`\#`	Номер команды (считает все команды с момента запуска shell)
`\$`	Отображает `$` для обычного пользователя и `#` для root
`\t`	Время в формате `HH:MM:SS`
`\T`	Время в формате `HH:MM` (12-часовой формат)
`\@`	Время в формате `hh:mm AM/PM`
`\d`	Дата (например: `Mon Aug 26`)
`\n`	Перенос строки
`\s`	Имя используемой оболочки (обычно `bash`)
`\v`	Версия Bash (например: `5.2`)
`\V`	Версия Bash с патч-уровнем
`\j`	Количество фоновых задач, запущенных из shell
`\!`	Номер команды в истории
`\#`	Номер команды (считает все команды в текущей сессии)
`\e` или `\033`	Escape-символ (для цветов/ANSI-последовательностей)

Пример: PS1="\u@\h:\w\$ "

PATH

Просмотр последовательно всех путей.

Добавление в пути: PATH=$PATH":/home/you/bin"

Для ускорения поиска пути к приложениям хранятся в хэш таблице в каждой сессии.

hash -r очищает таблицу хэшей путей.

CDPATH

Аналог path для директорий поиска при использовании cd.

Переменные окружения

Вызванная программа не знает переменных консоли. Для видимости нужно экспортировать переменную в переменную окружения.

env	Список переменных окружения
export varnames	Может быть список через пробел
export varname=value command	Доступность переменной только для определенной команды

source .envfile догружает переменные из файла

Переменные по умолчанию

BASH_VERSION	Версия
BASHPID	
GROUPS	Группы текущего пользователя
HOSTNAME	Имя хоста
OSTYPE	
PWD	Текущая рабочая директория
RANDOM	Случайное число от 0 до 32767
UID	Идентификатор пользователя
SHELL	Полный путь к текущей консоли

Периферийные функции

sleep n	Пауза на n секунд
seq 1 10	Последовательность чисел от 1 до 10
date '+%Y-%m-%d %H:%M:%S'	Текущая дата и время
tail -f fname	Выводит изменения файла в текущую консоль

Revision #27
Created 26 August 2025 08:47:23 by Admin
Updated 21 January 2026 15:50:46 by Admin