**ВОПРОСЫ К ЗАЩИТЕ**

\*Сашенька

1. Нулевой символ(\0) - символ завершения строки, то есть после него ничего не выведется. Является аналогом “ ”.

Вообще о спецсимволах в UNIX:

| Символ | Назначение |
| --- | --- |
| ; | Разделитель команд |
| & | Выполнение команд в фоновом режиме |
| () | Группирование команд |
| | | Канал (конвеер) |
| < > & | Знаки переадресации |
| \* ? [ ] ~ + - @ ! | Метасимволы подстановки имен файлов |
| " ' \ | Служат для заключения в кавычки других символов |
| ` | Подстановка команд |
| $ | Подстановка переменных (команд или арифметиче­ских выражений) |
| # | Обозначает начало комментария, продолжающегося до конца строки |
| Знаки пробела, табуляции | Разделители строк и новой строки |

? - обозначает "подстановочный" знак, формирует шаблон, который соответствует диапазону имен файлов, а не одному.

\* - "подстановочный" знак звездочка (\*) для обозначения любой *последовательности* символов, включая *отсутствие символов*.

[] - соответствует диапазону символов, которые надо искать.

; - разделитель команд оболочки.

~ - сокращение домашнего каталога.

1. В чём разница между различными кавычками? При обращении к переменным, желательно использовать двойные кавычки. Это позволит не интерпретировать специальные символы, содержащиеся в именах переменных, за исключением символов $, ` ( обратная кавычка) и \ ( обратный слэш ).

То что символ $ является исключением, позволяет производить подстановку переменных в строке. Кроме вышесказанного, двойные кавычки используются для предотвращения разбиения строки на слова.

Заключать в кавычки аргументы команды echo, нужно только если разбиение вывода на слова, вызывает какие-то трудности.

Обратные кавычки ` ` подставляют результат команды, которая написана между ними.(В одну строку).

s340384@helios:/home/s340384$ echo 'a\\nb'

a\nb

s340384@helios:/home/s340384$ echo "a\\nb"

a

b

Символ обратного слэша \ позволяет использовать управляющие символы без их интерпретации командной оболочкой; процедура добавления данного символа перед управляющими символами называется экранированием символов.

1. Как создать файл с именем fi"le

Чтобы создать файл с именем fi”le, просто заключим название в одиночные кавычки, наоборот тоже работает.

1. **echo - выводить строку текста в терминал**

**$ echo опции строка**

**-n** - не выводить перевод строки;

**-e** - включить поддержку вывода Escape последовательностей;

**-E** - отключить интерпретацию Escape последовательностей.

Это все опции, если включена опция -e, то вы можете использовать такие Escape последовательности для вставки специальных символов:

* **/c** - удалить перевод строки;
* **/t** - горизонтальная табуляция;
* **/v** - вертикальная табуляция;
* /b - удалить предыдущий символ;
* **/n** - перевод строки;
* **/r** - символ возврата каретки в начало строки

1. cd(change directory) - cмена директории.

**$ cd опции папка\_назначения**

* **-P** - позволяет следовать по символическим ссылкам перед тем, как будут обработаны все переходы "..";
* **-L** - переходит по символическим ссылкам только после того, как были обработаны "..";
* **-e** - если папку, в которую нужно перейти не удалось найти - выдает ошибку.

Результат выполнения:

cd / - возврат в корневую директорию(в той, где home)

cd lab0 - переход в директорию lab0, если она есть в текущей(в данном случае ошибка)

cd - переход в домашнюю директорию(/home/s339...)

cd lab0 - переход в директорию lab0, если она есть в текущей(в данном случае переходит в lab0)

cd ../../././../../ - переход в корневую директорию из lab0(./ - ничего не делает, ловушка джовушкера)

1. **Жесткая** **ссылка** не может указывать на файл **в** **другой** **файловой** **системе**. Например, **невозможно** **создать** на **жестком** диске **жесткую** **ссылку** на файл, расположенный на дискете. Причина этого ограничения в том, что номер индексного дескриптора уникален только в рамках одной **файловой** **системы**.

Линукс автоматически создаёт 2+n жёстких ссылок(первая на саму директорию и вторую на его директорию)

\*Митька

1. Какие аттрибуты времени у файла

**mtime** (время модификации) отображается в команде ls -l по-умолчанию

**atime** Время последнего доступа к файлу/каталогу

**ctime-** Время последнего изменения атрибутов файла/каталога (изменений в inode)

ls -l -ctime

**$ ls опции /путь/к/папке**

* **-a** - отображать все файлы, включая скрытые, это те, перед именем которых стоит точка;
* **-A** - не отображать ссылку на текущую папку и корневую папку . и ..;
* **--author** - выводить создателя файла в режиме подробного списка;
* **-b** - выводить Escape последовательности вместо непечатаемых символов;
* **--block-size** - выводить размер каталога или файла в определенной единице измерения, например, мегабайтах, гигабайтах или килобайтах;
* **-B** - не выводить резервные копии, их имена начинаются с ~;
* **-c** - сортировать файлы по времени модификации или создания, сначала будут выведены новые файлы;
* **-C** - выводить колонками;
* **--color** - включить цветной режим вывода, автоматически активирована во многих дистрибутивах;
* **-d** - выводить только директории, без их содержимого, полезно при рекурсивном выводе;
* **-D** - использовать режим вывода, совместимый с Emacs;
* **-f** - не сортировать;
* **-F** - показывать тип объекта, к каждому объекту будет добавлен один из специализированных символов \*/=>@|;
* **--full-time** - показывать подробную информацию, плюс вся информация о времени в формате ISO;
* **-g** - показывать подробную информацию, но кроме владельца файла;
* **--group-directories-first** - сначала отображать директории, а уже потом файлы;
* **-G** - не выводить имена групп;
* **-h** - выводить размеры папок в удобном для чтения формате;
* **-H** - открывать символические ссылки при рекурсивном использовании;
* **--hide** - не отображать файлы, которые начинаются с указанного символа;
* **-i** - отображать номер индекса inode, в которой хранится этот файл;
* **-l** - выводить подробный список, в котором будет отображаться владелец, группа, дата создания, размер и другие параметры;
* **-L** - для символических ссылок отображать информацию о файле, на который они ссылаются;
* **-m** - разделять элементы списка запятой;
* **-n** - выводить UID и GID вместо имени и группы пользователя;
* **-N** - выводить имена как есть, не обрабатывать контролирующие последовательности;
* **-Q** - брать имена папок и файлов в кавычки;
* **-r** - обратный порядок сортировки;
* **-R** - рекурсивно отображать содержимое поддиректорий;
* **-s** - выводить размер файла в блоках;
* **-S** - сортировать по размеру, сначала большие;
* **-t** - сортировать по времени последней модификации;
* **-u** - сортировать по времени последнего доступа;
* **-U** - не сортировать;
* **-X** - сортировать по алфавиту;
* **-Z** - отображать информацию о расширениях SELinux;
* **-1** - отображать один файл на одну строку.

2)mkdir -p - создание вложенных директорий.

3)echo и пробелы

s340384@helios:/home/s340384$ echo "a b"

a b

s340384@helios:/home/s340384$ echo a b

a b

4)создать файл, принадлежащий группе P3108

touch file

chgrp P3108 file

5)ls -r vs ls -R

s340384@helios:/home/s340384$ ls

lab\_0\_1.sh lab\_0\_4.sh Lab\_1

lab\_0\_2.sh lab\_0\_5.sh lab0

lab\_0\_3\_wrong.sh lab\_0\_delete\_all.sh login.sql

lab\_0\_3.sh lab\_0.sh mnt

s340384@helios:/home/s340384$ ls -r

mnt lab\_0.sh lab\_0\_3.sh

login.sql lab\_0\_delete\_all.sh lab\_0\_3\_wrong.sh

lab0 lab\_0\_5.sh lab\_0\_2.sh

Lab\_1 lab\_0\_4.sh lab\_0\_1

6) создать жёсткую ссылку на директорию

s340384@helios:/home/s340384$ ln test testln

ln: test is a directory

7)**ln опции файл\_источник файл\_ссылки**

Рассмотрим опции утилиты:

* **-d** - разрешить создавать жесткие ссылки для директорий суперпользователю;
* **-f** - удалять существующие ссылки;
* **-i** - спрашивать нужно ли удалять существующие ссылки;
* **-P** - создать жесткую ссылку (по умолчанию);
* **-r** - создать символическую ссылку с относительным путем к файлу;
* **-s** - создать символическую ссылку.

8)grep-нуть ошибки предыдущей команды