

Раздел «Язык Си» . Consol :

- Консольный режим

Консольный режим

Несколько основных команд

Команда	Назначение
pwd	указывает полное название текущего каталога (в которой работаем)
ls	показывает список файлов в текущем каталоге

Внимание! Знак > перед командой pwd набирать не нужно. Это приглашение (command prompt) в консоли. Оно разное в разных консолях, и его может быть другое приглашение на ввод. Далее в примерах команд применяемых в консоли, будет использоваться приглашение на ввод следующей строки – то, что напечатала команда или программа.

Задание: Запустите консоль в той операционной системе (ОС), где вы выполняете задание. Выполните команду pwd. Что она напечатала?

Файлы и каталоги

С точки зрения Unix все данные помещаются в файлы. Каталоги также являются файлами. В них – список файлов и других каталогов, которые размещены в каталоге. Кстати, один и тот же файл может одновременно находиться в нескольких каталогах и иметь разные имена. Unix рассматривает различные физические устройства: диски, флеш-карты, сетевые подключения и т.д. как файлы.

Кроме пользовательских файлов, существуют системные служебные файлы. По умолчанию, обычный пользователь не имеет к ним доступа (это определяется настройками ОС). Все каталоги размещены в одном корневом каталоге, обозначается /. В названиях файлов Unix различает строчные и прописные буквы на любом языке.

Как правило, пользовательские "домашние" каталоги находятся в каталоге home, который, в свою очередь, расположен в корневом каталоге. Имя каталога может быть **ПОЛНЫМ** (абсолютным) и **ОТНОСИТЕЛЬНЫМ**. Если мы работаем в корневом каталоге, то home – **относительное** имя (относительно текущего каталога, то есть *сейчас* относительно корневого каталога), а /home – **полное** имя.

Имя файла (каталога) еще называют **путь**. Путь может быть абсолютным или относительным. Абсолютный путь начинается с /.

Допустим есть пользователь с именем student, который создал себе рабочий каталог workspace и поместил в него файл myprog.c

Файл или каталог	Что это значит
myprog.c	относительное имя, если пользователь находится в каталоге workspace
workspace/myprog	имя, относительно каталога student
/home/student/workspace/myprog	полное имя файла
./	обозначение ТЕКУЩЕГО каталога
./myprog.c	явное обозначение файла в текущем каталоге
../	обозначение каталога выше

Поиск

Поиск

Раздел «Язык Си»

Главная

Зачем учить C?

Определения

Инструменты:

Поиск

Изменения

Index

Статистика

Разделы

Информация

Алгоритмы

Язык Си

Язык Ruby

Язык

Ассемблера

EI Judge

Парадигмы

Образование

Сети

Objective C

Logon>>

Некоторые команды для работы с файлами и папками (размещение, перенос, переименование)

При работе с файлами и каталогами их необходимо: создавать, перемещать, копировать в другое место, удалять и т.д.

Для исполнения команды, ее нужно набрать на консоли и нажать ENTER.

ls

ls каталог

Просмотр указанного каталога. Если каталог в качестве *аргумента* команды не указан, то подразумевается текущий каталог.

Пусть в каталоге /home находится только каталог student. В нем находится папка work и файл my.c.

Текущий каталог /home. Выполним следующие действия

- Просмотр содержимого текущего каталога (это /home)

```
> ls  
student
```

- просмотр содержимого каталога student

```
> ls student  
work my.c
```

mkdir каталог

Создать папку каталог .

- просмотр текущего (пустого) каталога;
- создание каталога (папки) myWorkFolder
- просмотр текущего каталога (в нем оказывается только что созданный каталог myWorkFolder)

```
> ls
```

```
> mkdir myWorkFolder  
> ls  
myWorkFolder
```

cd

cd каталог

Перейти в каталог . Если каталог в качестве аргумента не указан, перейти в домашний.

- просмотр текущего каталога
- переход в каталог myWorkFolder
- просмотр нового текущего каталога (myWorkFolder)
- переход на каталог выше (обратно)
- Просмотр каталога
- Переход в каталог /home
- Переход в *домашний каталог*

```
> ls  
myWorkFolder  
> cd myWorkFolder  
> ls  
myprog.c  
> cd ..  
> ls  
myWorkFolder  
> cd /home  
> cd ~
```

ср откуда куда

Копировать файл *откуда* в *куда*

- Копирование файла myprog.c в файл yourprog.c и просмотр (есть оба файла).

```
>ср myprog.c yourprog.c
>ls
myprog.c yourprog.c
```

- Копирование файла из myWorkFolder в текущий каталог. В текущем каталоге появится файл с таким же именем.

```
>ср myWorkFolder/myprog.c .
```

- Копирование файла 'otherfile' в каталог myWorkFolder. В каталоге myWorkFolder появится файл с таким же именем

```
>ср otherfile myWorkFolder/
```

mv откуда куда

Переместить файл *откуда* в *куда* . Файл (каталог) *откуда* будет удален. Используется для переименования и перемещения файлов (каталогов).

- Переименование файла myprog.c в файл yourprog.c и просмотр (файла с именем myprog.c не стало)

```
>ls
myprog.c
>mv myprog.c yourprog.c
>ls
yourprog.c
```

- Перемещение файла из myWorkFolder в текущий каталог. В текущем каталоге появится файл с таким же именем, в каталоге myWorkFolder будет удален.

```
>mv myWorkFolder/myprog.c .
```

- Перемещение файла otherfile в каталог myWorkFolder. В каталоге myWorkFolder появится файл с таким же именем, в текущем будет удален.

```
>mv otherfile myWorkFolder/
```

Создание и изменение файлов.

Все файлы создаются специальными программами. Мы будем работать с текстовыми файлами (это – написанные программы и данные) и запускаемыми файлами.

Запускаемые файлы мы будем получать из текстовых файлов-программ. А сами программы и другие тексты пишутся в тестовых редакторах, которых великое множество.

Текстовые редакторы разделяются на интерактивные и неинтерактивные.

Неинтерактивные редакторы

cat

cat – простейший неинтерактивный текстовый редактор.

```
> cat myprog.c
#include <stdio.h>
int main()
{
    int a;
    /* это комментарий */

    printf("Hello");
}
```

Просмотр содержимого файла 'myprog.c'. Вывод происходит на экран.

Если 'cat' не указать файл для вывода, он будет ожидать ввод с клавиатуры и сразу все печатать на экран. Символ '>' (перенаправление вывода) позволяет перенаправить вывод с экрана в файл. При этом либо создается новый файл, либо полностью перезаписывается содержимое уже существующего файла.

Создание нового файла 'newfile.txt'.

В файл будет записан весь текст, который набирается на клавиатуре. Чтобы закончить ввод и закрыть файл, нужно ввести новую строку (**ENTER**) и нажать **CTRL+D**

Символы '>>' позволяют добавить вывод в конец уже существующего файла без удаления записанной ранее информации.

```
>cat>newfile.txt
1234567
>ls
myprog.c newfile.txt
>cat>>newfile.txt
abcdef
>cat newfile.txt
1234567
abcdef
```

Редактор vim

Для редактирования текстов существует консольный редактор vim.

vim может находиться в:

- нормальном режиме,
- режиме ввода команд,
- режиме ввода текста,
- визуальном режиме.

Запуск vim:

```
>vim mytext
```

vim выполняет команды клавиатуры: сочетание различных клавиш и командных строк, набранных на клавиатуре. Сразу после запуска пользователь оказывается в *нормальном режиме*: в котором можно ставить метки, копировать, удалять, вставлять текст их буферов и т.д

Для ввода текста необходимо перейти в режим ввода текста. Самый простой способ – клавиша "i".

Вернуться в нормальный режим – клавиша ESC

Здесь точно нужно видео

vim: основные команды в нормальном режиме (просмотр)

Для перехода в нормальный режим нажмите клавишу ESC.

Клавиши или их сочетания	Что делает
h,j,k,l	перемещение в разные стороны или стрелки (Когда vi был написан (в 70-х годах прошлого века), стрелки не всегда были доступны . Поэтому h, j, k, и l были выбраны как команды для перемещения влево,вниз,вверх и вправо соответственно.)
i	режим вставки (insert)
I	добавление в начало строки
a	режим добавления (append)
A	добавление в конец строки
o	вставить строку за текущей (out)
O	вставить строку перед текущей

Клавиши или их сочетания	Что делает
R	писать поверх (replace)
u	отмена предыдущего действия (undo)
CTR-R	отмена отмены предыдущего действия (redo)
dd	удалить строку и сохранить в буфер (delete)
yy	копировать строку в буфер (copy)
p	вставить из буфера обмена (paste)
{n}d	удалить n+1 строку
{n}y	скопировать n+1 строку
ESC	перейти в режим просмотра
DEL	удалить следующий символ
:{n}	перейти на строку n
:e myfile	открыть для редактирования файл (edit)
:w myfile	записать изменения (write)
:wa	сохранить изменения во всех открытых файлах (write all)
:q	выйти из редактора (quit)
:q!	выйти из редактора без сохранения
:cd [path]	перейти в другой каталог
:!команда	выполнить команду – man, gcc, и так далее
CTR+u, CTR+d	Page Up / Page Down
CTR+y, CTR+e	Перемотка вверх/вниз без движения курсора

vim: Маркеры

Для быстрого перехода по нужным вам позициям в файле используют маркеры. Работать с маркерами можно в нормальном режиме.

Клавиши или их сочетания	Что делает
ma	установить локальный маркер a (имя маркера любая буква) на позицию, в которой стоит курсор
mB	установить глобальный маркер B
`\${\$c	перейти к локальному маркеру c (при использовании маркеров для выделения текста при копировании, удалении и т.д выделяется весь текст от текущего положения курсора до маркера)
`\${\$c	переход к строке, на которой установлен маркер (перед именем маркера клавиша SHIFT+~, обратная кавычка)
:marks	просмотр маркеров

vim: Поиск и замена

Специальные символы для поиска	Что означает
/n	перенос строки
/t	табуляция
/s	пробел

Специальные символы для поиска	Что означает
/<	слово должно начинаться с набора символов, который стоит за этими знаками
/>	слово должно заканчиваться набором символов который стоит перед этими знаками
*	символ, после которого стоит * может встречаться сколько угодно раз (от 0)
.	точка означает "любой символ"
/ {n}	последовательность символов перед этим должна встречаться не менее n раз
[\$набор символов\$]\$	в строке должен встречаться какой-нибудь из перечисленных в скобках символов.

-- TatyanaOvsyannikova2011 - 09 Sep 2019

(с) Материалы раздела "Язык Си" публикуются под лицензией GNU Free Documentation License.

Командная строка или клавиша	Что делает
/текст	поиск строки текста в файле сверху вниз
?текст	поиск строки текста в файле снизу вверх
n	следующее совпадение
N	предыдущее совпадение
:%s/текст1/текст2/g	замена строки текст1 на текст2 во всем файле