

Что происходит после включения ПК

01

Срабатывает Basic Input/Output System (BIOS)

02

The Power-On-Self-Test (POST)

03

Чтение загрузочного сектора диска

04

Загрузка ОС

05

Запуск системных утилит, окружения

Упрощенная схема
логической структуры компьютера

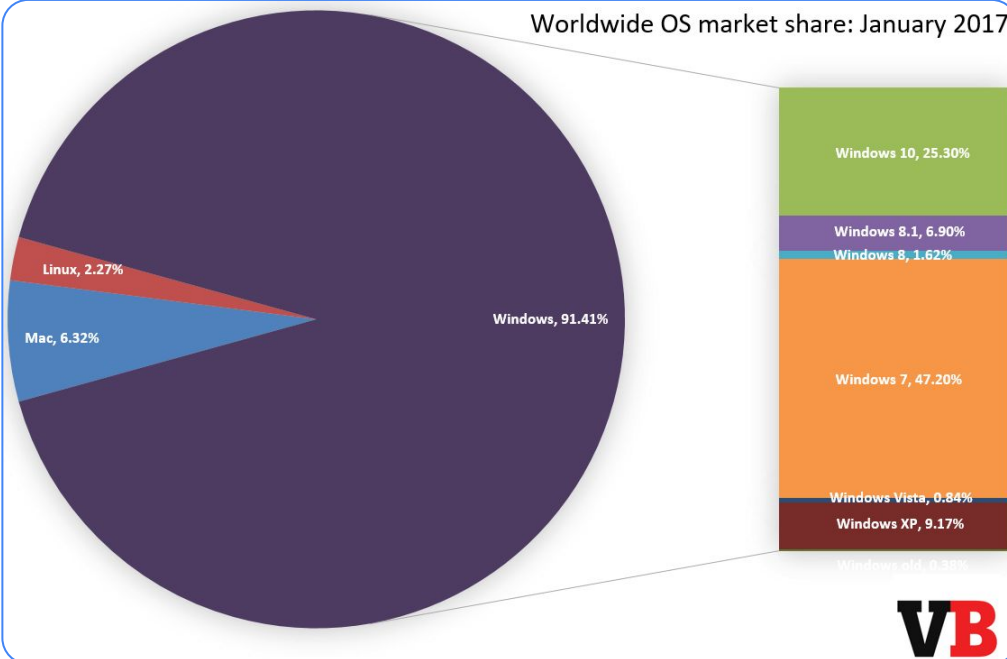
<http://inf1.info>



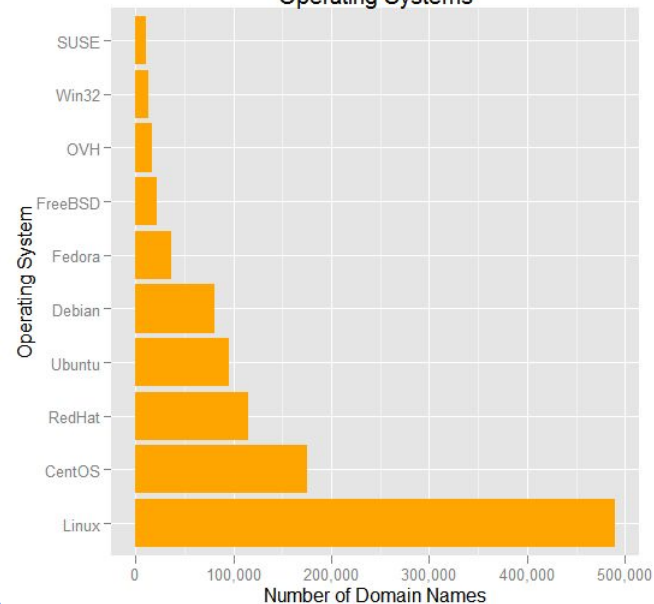
Операционная система

Распространенность ОС

Worldwide OS market share: January 2017



Operating Systems

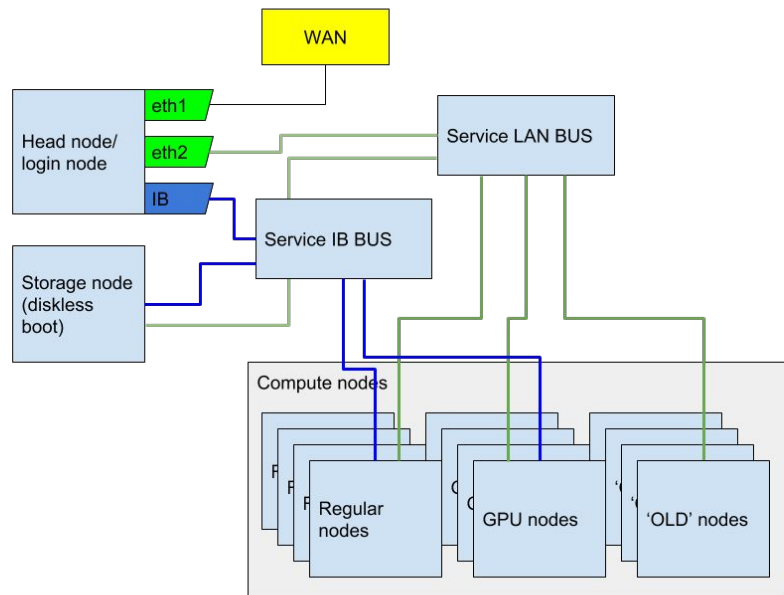


Рабочие станции, сервера, кластеры, системы хранения данных.

Специализированные компьютеры



VS

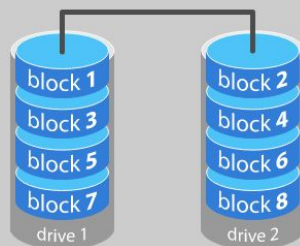


Системы хранения данных.

СХД - специализированный компьютер с большим хранилищем.

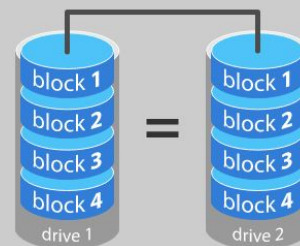
Основная задача - обеспечить надежность и доступность данных

RAID 0
striping



prepressure.com

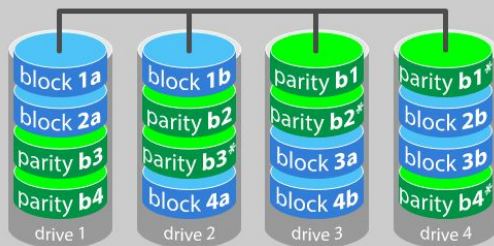
RAID 1
mirroring



prepressure.com

RAID 6

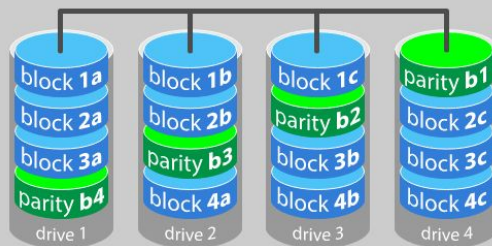
striping with dual parity across drives



prepressure.com

RAID 5

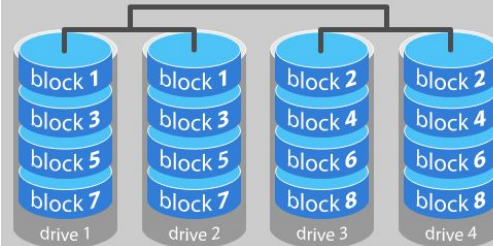
striping with parity across drives



prepressure.com

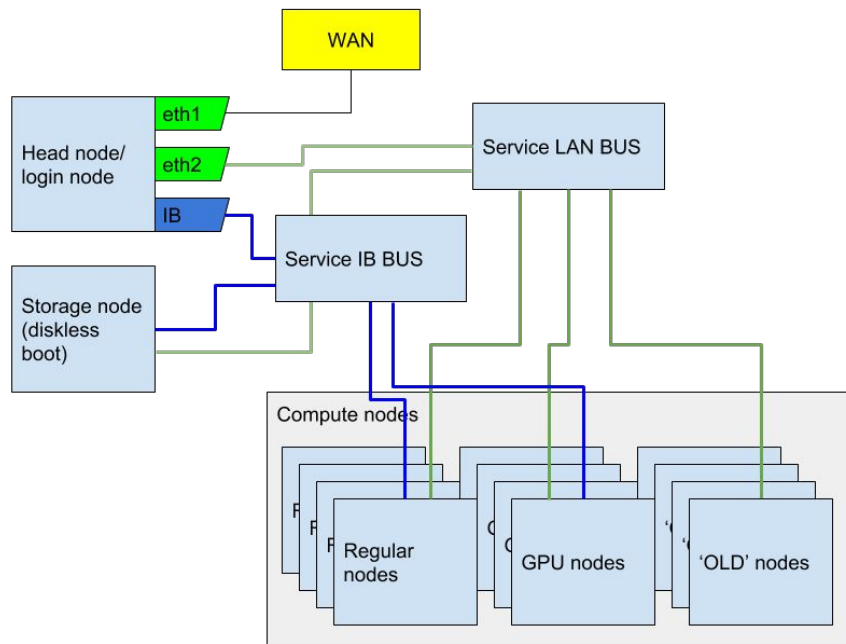
RAID 1+0

mirroring + striping



prepressure.com

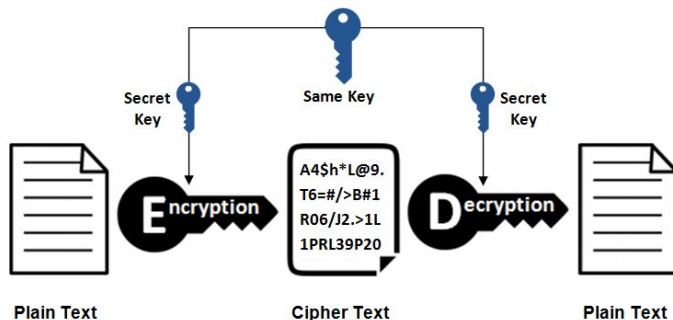
Вычислительный кластер. Newton



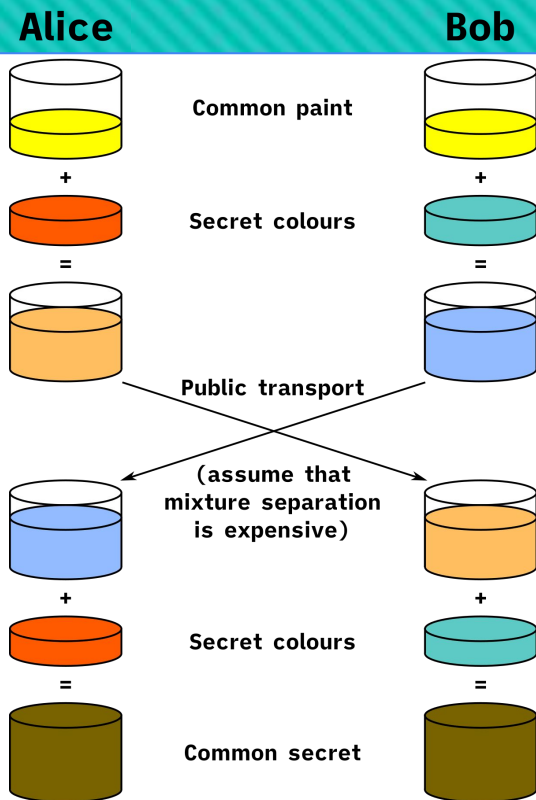
SSH. Криптоключи.

- Для доступа к компьютеру используют Secure Shell.
- Соединение всегда зашифровано.
- Linux и Mac содержат ssh клиент в дистрибутиве.
- Для windows - Putty.
- SSH позволяет туннелировать трафик, передавать файлы

Symmetric Encryption

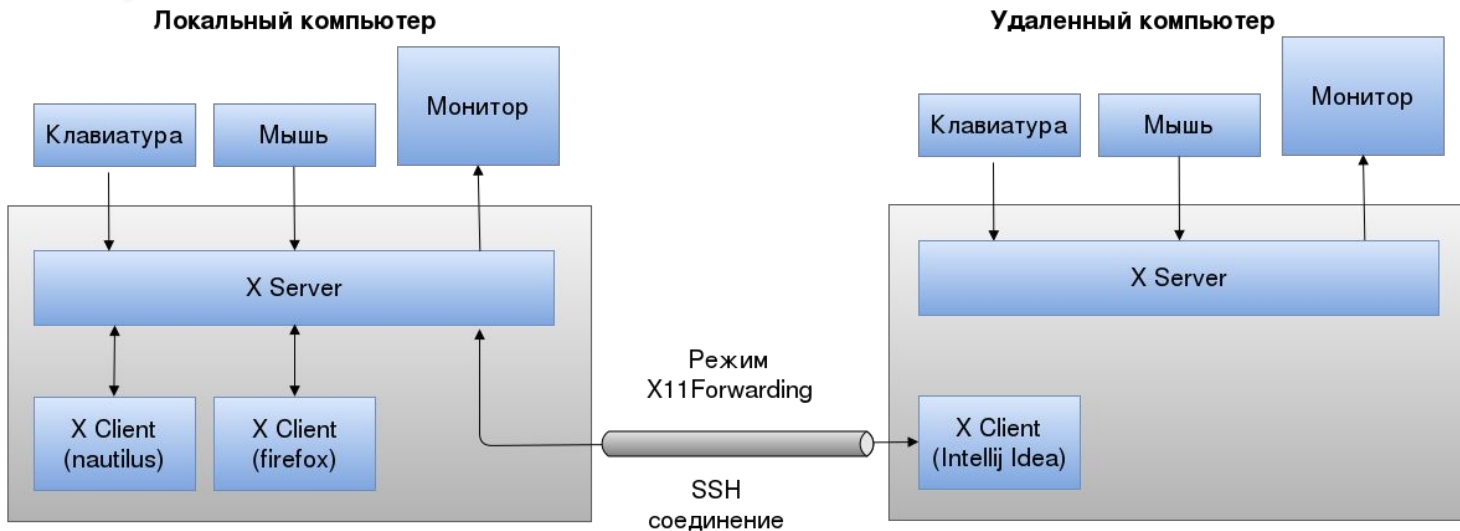


Diffie-Hellman Key Exchange



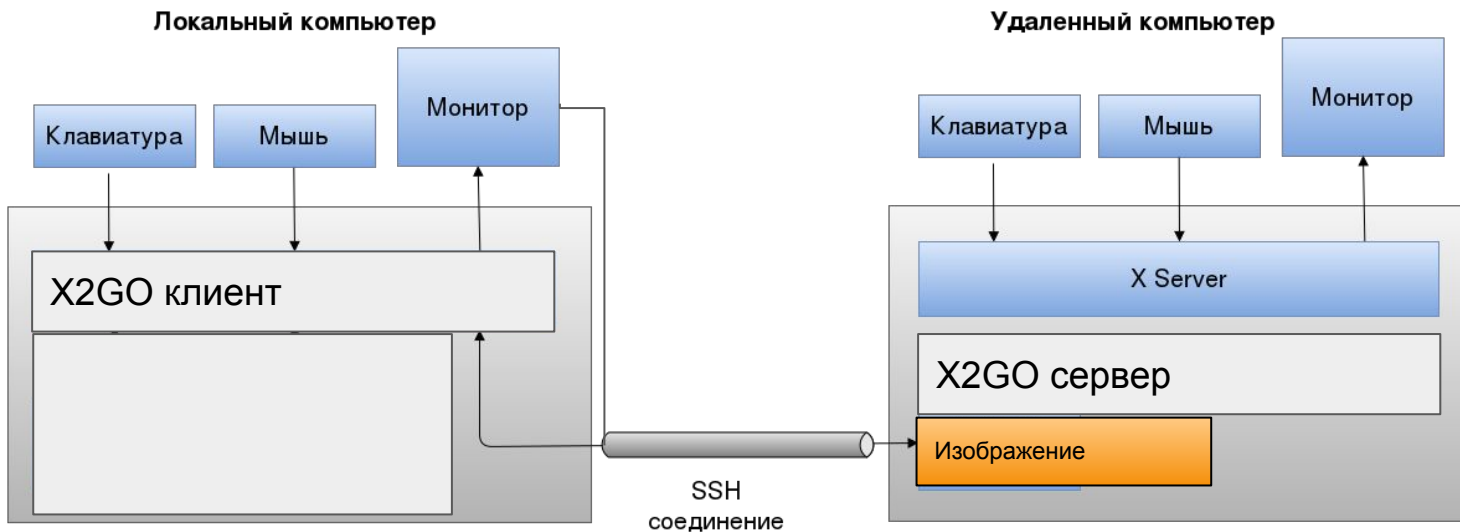
X11 forwarding, X2GO

- X Window System — оконная система, обеспечивающая стандартные инструменты и протоколы для построения графического интерфейса пользователя. Используется в UNIX-подобных ОС (не всегда).
- X11Forwarding перенаправляет трафик удаленного X Client'a локальному X Server'у.
- Окно приложения отображается на локальном рабочем столе, пока вся логика выполняется на удаленном компьютере.



X11 forwarding, X2GO

- X2GO - решает проблемы совместимости



Основные команды BASH

Directory Operations

<code>pwd</code>	Show current directory
<code>mkdir dir</code>	Make directory <i>dir</i>
<code>cd dir</code>	Change directory to <i>dir</i>
<code>cd ..</code>	Go up a directory
<code>ls</code>	List files

ls Options

<code>-a</code>	Show all (including hidden)
<code>-R</code>	Recursive list
<code>-r</code>	Reverse order
<code>-l</code>	Sort by last modified
<code>-S</code>	Sort by file size
<code>-l</code>	Long listing format
<code>-1</code>	One file per line
<code>-m</code>	Comma-separated output
<code>-Q</code>	Quoted output

Bash Shortcuts

<code>CTRL-c</code>	Stop current command
<code>CTRL-z</code>	Sleep program
<code>CTRL-a</code>	Go to start of line
<code>CTRL-e</code>	Go to end of line
<code>CTRL-u</code>	Cut from start of line
<code>CTRL-k</code>	Cut to end of line
<code>CTRL-r</code>	Search history
<code>!!</code>	Repeat last command
<code>!abc</code>	Run last command starting with <i>abc</i>
<code>!abc:p</code>	Print last command starting with <i>abc</i>
<code>!\$</code>	Last argument of previous command
<code>ALT-.</code>	Last argument of previous command
<code>!*</code>	All arguments of previous command
<code>^abc^123</code>	Run previous command, replacing <i>abc</i> with <i>123</i>

Bash Variables (cont)

<code>export NAME=value</code>	Set <i>\$NAME</i> to <i>value</i>
<code>\$PATH</code>	Executable search path
<code>\$HOME</code>	Home directory
<code>\$SHELL</code>	Current shell

Search Files

<code>grep pattern files</code>	Search for <i>pattern</i> in <i>files</i>
<code>grep -i</code>	Case insensitive search
<code>grep -r</code>	Recursive search
<code>grep -v</code>	Inverted search
<code>grep -o</code>	Show matched part of file only
<code>find /dir/ -name name*</code>	Find files starting with <i>name</i> in <i>dir</i>

Command Lists

<code>cmd1 ; cmd2</code>	Run <i>cmd1</i> then <i>cmd2</i>
<code>cmd1 && cmd2</code>	Run <i>cmd2</i> if <i>cmd1</i> is successful
<code>cmd1 cmd2</code>	Run <i>cmd2</i> if <i>cmd1</i> is not successful
<code>cmd &</code>	Run <i>cmd</i> in a subshell

Bash Commands

<code>uname -a</code>	Show system and kernel
<code>head -n1 /etc/issue</code>	Show distribution
<code>mount</code>	Show mounted filesystems
<code>date</code>	Show system date
<code>uptime</code>	Show uptime
<code>whoami</code>	Show your username
<code>man command</code>	Show manual for <i>command</i>

IO Redirection

<code>cmd < file</code>	Input of <i>cmd</i> from <i>file</i>
<code>cmd1 <(cmd2)</code>	Output of <i>cmd2</i> as file input to <i>cmd1</i>
<code>cmd > file</code>	Standard output (stdout) of <i>cmd</i> to <i>file</i>
<code>cmd > /dev/null</code>	Discard stdout of <i>cmd</i>
<code>cmd >> file</code>	Append stdout to <i>file</i>
<code>cmd 2> file</code>	Error output (stderr) of <i>cmd</i> to <i>file</i>

Pipes

<code>cmd1 cmd2</code>	stdout of <i>cmd1</i> to <i>cmd2</i>
<code>cmd1 & cmd2</code>	stderr of <i>cmd1</i> to <i>cmd2</i>

Файловая система

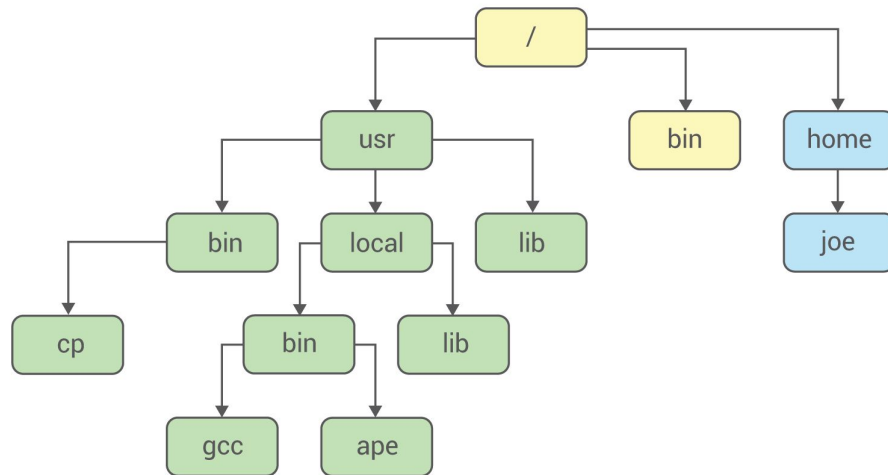
Рабочая директория - `pwd`

директория выше - `../`

текущая директория - `./`

визуальный файловый

менеджер - `mc`



Запуск программ в командной строке

- Запуск программ в командной строке: файл должен быть исполняемым (ll, chmod)
- При запуске необходимо указывать ПУТЬ к файлу.
 - либо полный путь - /home/user/script.sh
 - либо относительный - ./script.sh
- Запуск Python-скриптов осуществляется через интерпретатор Python

Переменные среды

```
vivek@nas01:~$ printenv
TERM=xterm-256color
SHELL=/bin/bash
XDG_SESSION_COOKIE=9ee90112ba2cb349f07bfe2f00002e46-1381581541.324726-906214463
SSH_CLIENT=192.168.1.6 60190 22
SSH_TTY=/dev/pts/1
USER=vivek
MAIL=/var/mail/vivek
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
PWD=/home/vivek
LANG=en_IN
SHLVL=1
HOME=/home/vivek
LANGUAGE=en_IN:en
LOGNAME=vivek
SSH_CONNECTION=192.168.1.6 60190 192.168.1.10 22
_=/usr/bin/printenv
vivek@nas01:~$
```

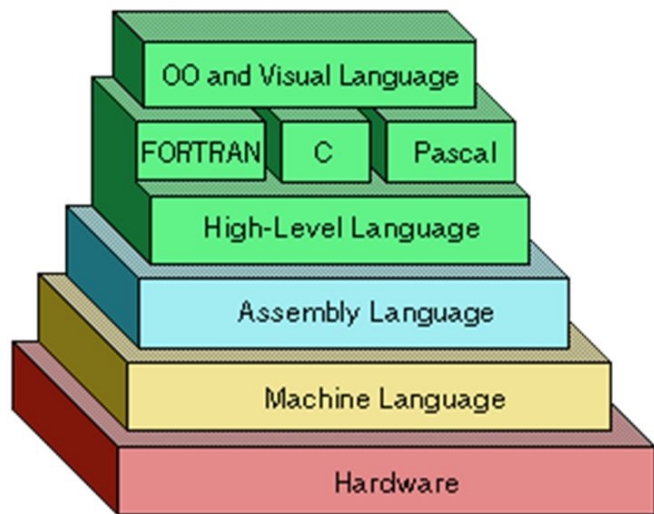
- PATH - путь к папкам с исполняемыми файлами
- LD_LIBRARY_PATH - путь к библиотекам
- PYTHONPATH - путь к модулям
- ...

Использование модулей, команда module

- module avail
- module list
- module load
- module unload

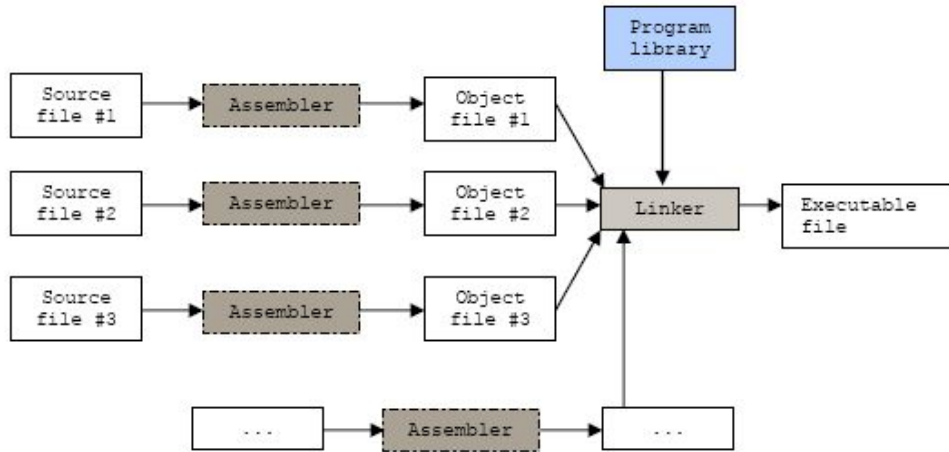
```
vivek@nas01:~$ printenv
TERM=xterm-256color
SHELL=/bin/bash
XDG_SESSION_COOKIE=9ee90112ba2cb349f07bfe2f00002e46-1381581541.324726-906214463
SSH_CLIENT=192.168.1.6 60190 22
SSH_TTY=/dev/pts/1
USER=vivek
MAIL=/var/mail/vivek
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
PWD=/home/vivek
LANG=en_IN
SHLVL=1
HOME=/home/vivek
LANGUAGE=en_IN:en
LOGNAME=vivek
SSH_CONNECTION=192.168.1.6 60190 192.168.1.10 22
_=/usr/bin/printenv
vivek@nas01:~$
```

Компиляторы, компиляция простейших программ



- cc - компилятор C
- g++ - компилятор C++
- make, cmake -
конфигураторы
процесса сборки

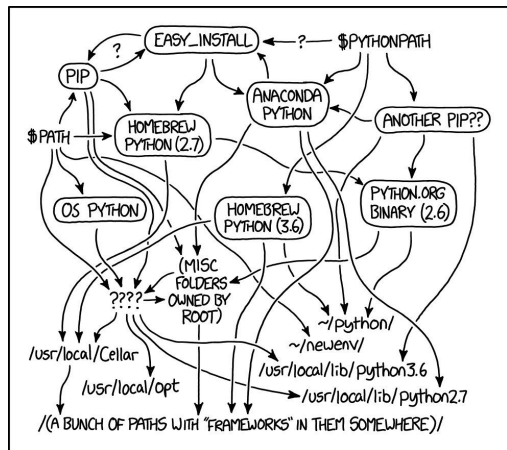
Компиляторы, компиляция простейших программ



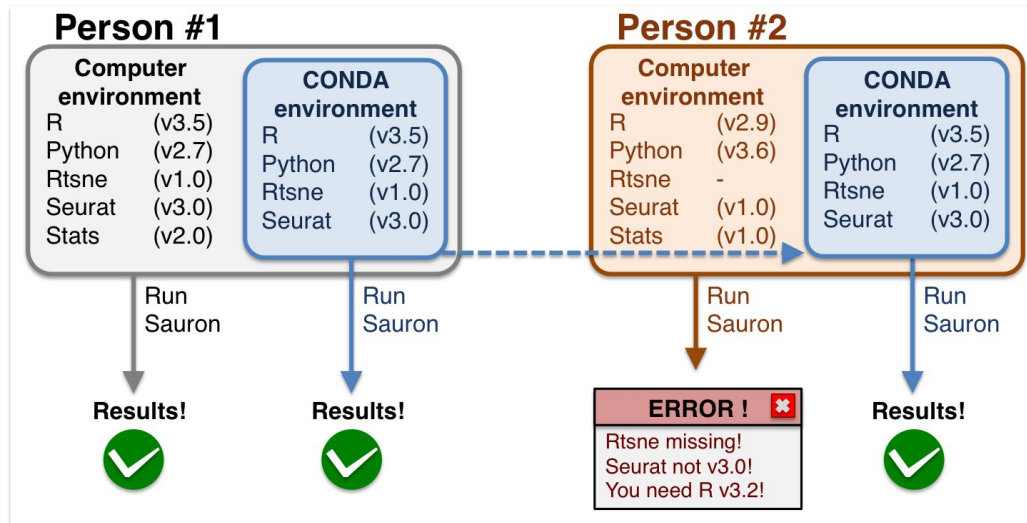
- cc - компилятор C
- c++ - компилятор C++
- make, cmake -
конфигураторы
процесса сборки

Менеджер пакетов conda, pip

Менеджер пакетов позволяет устанавливать модули нужной версии и воспроизводить “окружения”



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.



Менеджер пакетов conda, pip

Managing conda and anaconda

<code>conda info</code>	Verify conda is installed, check version #
<code>conda update conda</code>	Update conda package and environment manager to current version
<code>conda update anaconda</code>	Update the anaconda meta package (the library of packages ready to install with <code>conda</code> command)

Managing environments

<code>conda info --envs</code> or <code>conda info -e</code>	Get a list of all my environments, active environment shown with *
<code>conda create --name snowflakes biopython</code> or <code>conda create -n snowflakes biopython</code>	Create an environment and install program(s) in the environment (snowflakes) at the same time. <i>TIP: To avoid dependency conflicts, install all programs in your conda directory. You can specify a different path; see <code>conda create --help</code> for details.</i>
<code>source activate snowflakes</code> (Linux, OS X) <code>activate snowflakes</code> (Windows)	Activate the new environment to use it <i>TIP: Activate prepends the path to the snowflakes environment.</i>
<code>conda create -n bunnies python=3.4 astroid</code>	Create a new environment, specify Python version
<code>conda create -n flowers --clone snowflakes</code>	Make exact copy of an environment
<code>conda remove -n flowers --all</code>	Delete an environment
<code>conda env export > puppies.yml</code> <code>conda env create -f puppies.yml</code>	Save current environment to a file Load environment from a file

Managing Python

<code>conda search --full-name python</code> or <code>conda search -f python</code>	Check versions of Python available to install
<code>conda create -n snakes python=3.4</code>	Install different version of Python in new environment
<code>source activate snakes</code> (Linux, OS X) <code>activate snakes</code> (Windows)	Switch to the new environment that has a different version of Python <i>TIP: Activate prepends the path to the snakes environment.</i>

Managing .condarc configuration

<code>conda config --get</code>	Get all keys and values from my .condarc file
<code>conda config --get channels</code>	Get value of the key channels from .condarc file
<code>conda config --add channels pandas</code>	Add a new value to channels so conda looks for packages in this location

Git и GitHub.

Хранится история изменений файлов и их содержимого.
Diff.



```
36 src/material-examples/dialog-overview/dialog-overview-example.ts
... @@ -1,24 +1,46 @@
1 -import {Component} from '@angular/core';
2 -import {MdDialog} from '@angular/material';
3
4 /**
5  * @title Dialog Overview
6  */
7 @Component({
8   selector: 'dialog-overview-example',
9 -  templateUrl: 'dialog-overview-example.html',
10 })
11 export class DialogOverviewExample {
12
13   constructor(public dialog: MdDialog) {}
14
15 -  openDialog() {
16 -    this.dialog.open(DialogOverviewExampleDialog);
17
18 +import {Component, Inject} from '@angular/core';
19 +import {MdDialog, MdDialogRef, MD_DIALOG_DATA} from '@angular/material';
20
21 /**
22  * @title Dialog Overview
23  */
24 @Component({
25   selector: 'dialog-overview-example',
26 +  templateUrl: 'dialog-overview-example.html'
27 })
28 export class DialogOverviewExample {
29
30 +  animal: string;
31 +  name: string;
32
33 +  constructor(public dialog: MdDialog) {}
34
35 +  openDialog(): void {
36 +    let dialogRef = this.dialog.open(DialogOverviewExampleDialog, {
```

