

工具

yum——包管理器（类似应用商店）

认识生态

软件生态：面相对象（企业or学生or...）论坛 官方文档 软件....

开源是一种商业模式，可以于大公司绑定，更好的更低成本的维护自己的项目，获得补助，建立生态

命令

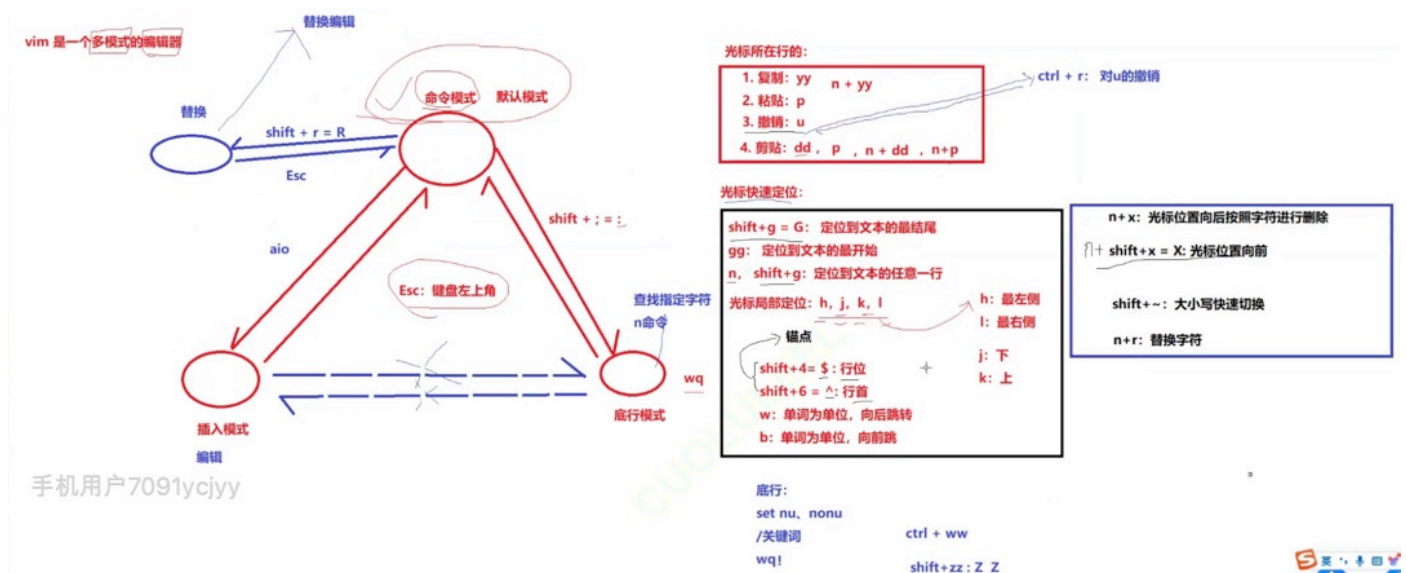
yum install -y.....

yum remove...

只能在root用户下下载

VIM GCC

命令模式命令



注释: ctrl + v hjkl(光标定位) shift+i(l) -> // -> esc

去注释: ctrl+v -> hjkl -> d

i a (光标向后1单位) o (光标换行) 进入插入模式

GCC

1. 预处理 (进行宏替换)

`gcc -E code.c -o code.i`

开始进行程序的翻译, 预处理做完就停下来

2. 编译 (生成汇编)

`gcc -S code.i -o code.s`

开始进行程序的翻译, 编译完成就停下来

3. 汇编 (生成机器可识别代码)

可重定位目标文件

`gcc -c code.c -o code.o`

开始进行程序的翻译, 汇编完成, 就停下

4. 连接 (生成可执行文件或库文件)

`gcc -o code code.o`

XXX.o 目标文件和标准库进行关联, 构建可执行程序

Esco

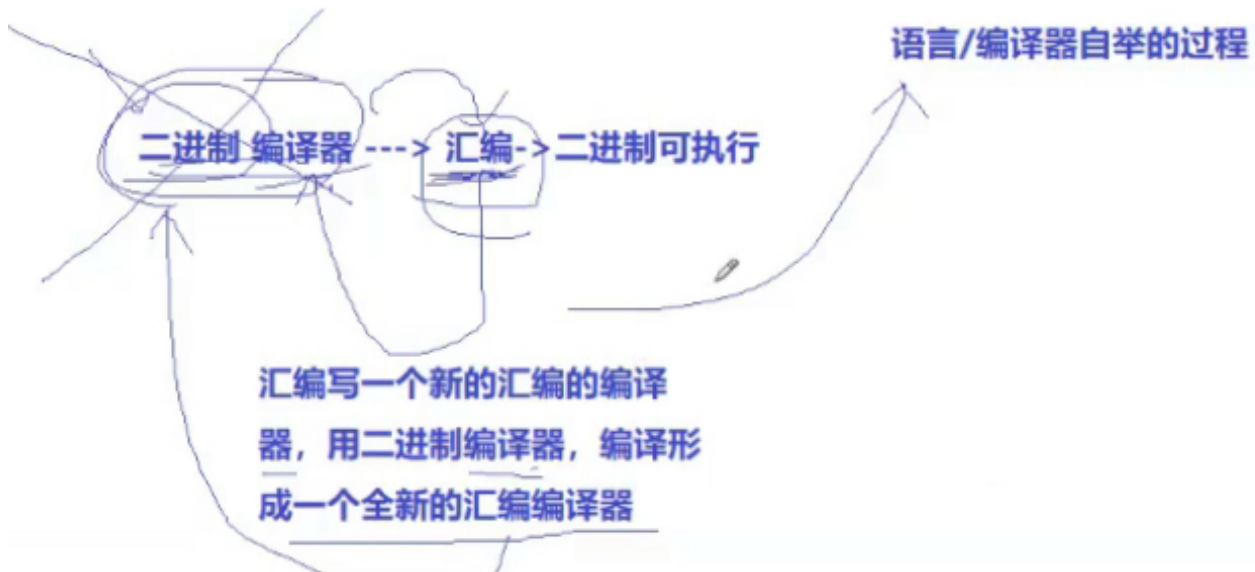
ldd获取文件依赖的库

```
[jym@VM-8-3-centos ~]$ ldd test
linux-vdso.so.1 => (0x00007ffff4cff3000)
libc.so.6 => /lib64/libc.so.6 (0x00007f6dcce7b000)
/lib64/ld-linux-x86-64.so.2 (0x00007f6dcd249000)
..
```

GCC的生态

编译器自举的过程

先有语言，还是先有编译器 手机用户7091ycjyy



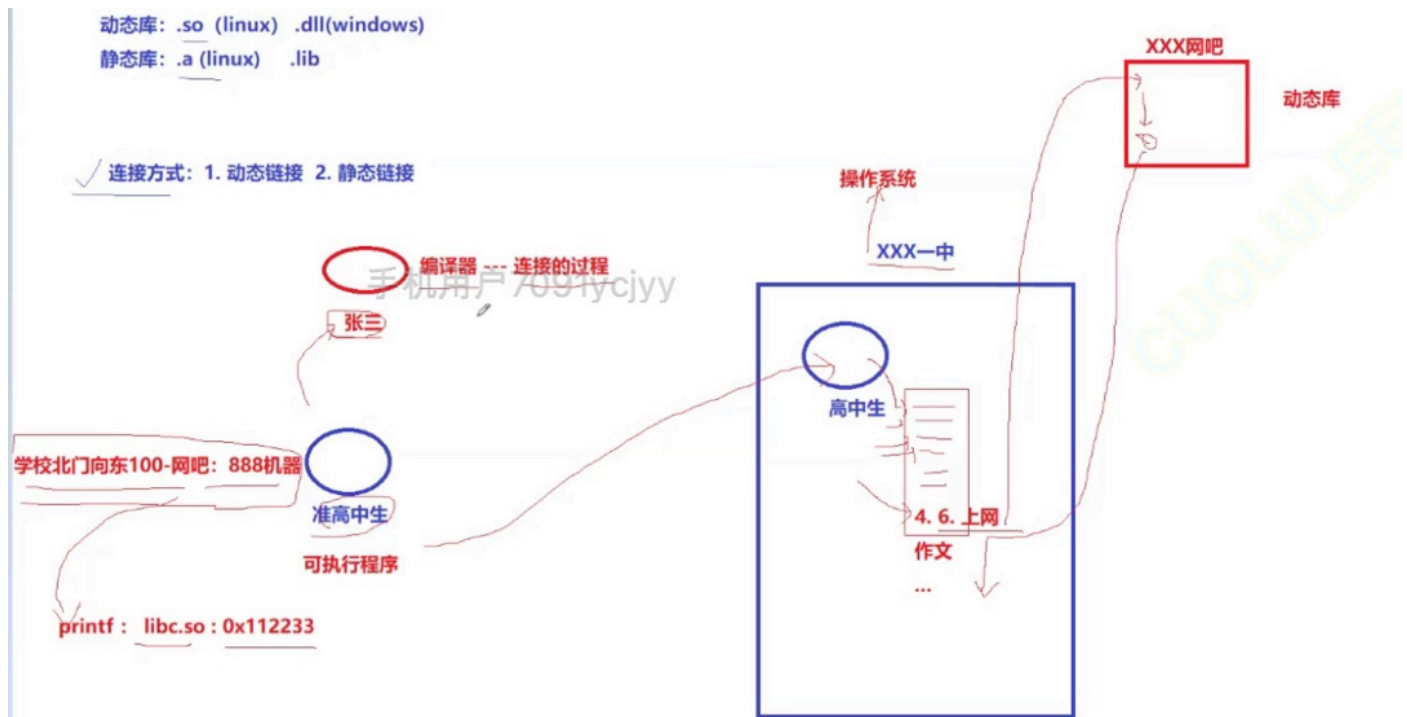
动态库静态库

动态库：.so(Linux) .dll(windows)

静态库：.a(Linux) .lib(windows)

动态链接：

运行时找



一旦动态库缺失，所有程序都无法运行

比较节省资源 可执行程序体积小

静态链接

编译时“替换”

静态链接：

把我们要访问的方法实现，拷贝到

我们自己的可执行程序中！

比较浪费资源，可执行程序体积比较大

不依赖其他库，库丢失也无所谓，程序照样能运行