

第二章 Linux常用命令

主要内容



- ✓ 基本命令的使用
- ✓ 文件操作的命令
- ✓ 目录操作的命令
- ✓ 查看帮助的命令
- ✓ 进程管理的命令
- ✓ 文件压缩/解压缩命令

使用命令



✓ 文件的命名规则

- ✓ 文件名可以由ASCII字符组合构成，除了 “/” 之外，所有字符都合法
- ✓ 有些字符最好不用，容易引起歧义。如：@ # & () 空格，制表符等
- ✓ 以 (.) 开头的文件名是隐藏文件
- ✓ 文件名区分大小写

使用命令

✓ 命令格式

✓ 例子

`ls -l`

`ls -l file1 file2`

`ls -la /root`

✓ 说明

- ✓ 命令、选项、参数之间要用空格隔开
- ✓ 当有多个选项时，可以写在一起
- ✓ 命令名必须是小写



简单命令

✓ who命令

- ✓ 列出所有正在使用系统的用户相关信息

who

//列出所有使用系统的用户、终端名、登陆时间

who am i

//列出使用该命令的当前用户的相关信息

whoami

//只列出当前用户名

简单命令

✓ echo命令

- ✓ 将命令行中的参数显示到屏幕上
- ✓ 常用于shell脚本中作为输出提示
- ✓ 例子

echo hello

echo \$SHELL

echo 'hello world'

简单命令

✓ date命令

✓ 显示或设置系统的日期和时间

✓ 例子

date

date 030412302018

date --set "03/07/2019 10:19"

✓ 说明

将时间更改为2018年3月4日12点30分的时间显示格式:

[MMDDhhmmYY]

简单命令

✓ cal 命令

✓ 显示日历

cal

//不带任何参数，则显示本月的日历

cal 2011

//带1个参数，显示本年的日历（12个月）

cal 10 2011

//带2个参数，显示某年某月的日历

简单命令

✓ clear 命令

- ✓ 清除屏幕上的信息

✓ passwd命令

- ✓ 用户修改密码

文件类型

✓ 普通文件

- ✓ 文本文件
- ✓ 数据文件
- ✓ 可执行文件

✓ 目录

- ✓ . 表示目录本身
- ✓ .. 表示该目录的父目录

✓ 设备文件

✓ 符号链接文件

文件操作命令

- ✓ 文件显示
- ✓ 查找、排序
- ✓ 复制、删除和移动
- ✓ 文件内容统计

文件显示命令

✓ cat命令—concatenate and display files

✓ 格式

cat [选项] 文件名

✓ 常见选项

-b 对所有非空行进行编号

-n 对所有行进行编号

✓ 例子

cat text //显示文件text

cat file1 file2 //显示file1, file2的内容

cat file1 file2>file3 //把file1, file2的内容连接起来, 合并为file3

文件显示命令

✓ more命令 --分页显示文件内容

✓ 格式

more [选项] 文件名

✓ 选项

-num 一次显示多少行

+num 从第num行开始显示

✓ 例子

more file1

more -10 file1

cat file1 | more

文件显示命令

✓ more命令 -- 分页显示文件内容

✓ 操作

space键：翻页

enter键：下一行

q/Q：中途退出more命令

h：显示帮助信息

文件显示命令

✓ less命令--分页显示文件的内容

✓ 格式

less [选项] 文件名

✓ 例子

less file1

✓ 说明

- ✓ less命令除可按空格键向下显示文件外，还可以利用pageup、pagedown、上下箭头来卷动文件
- ✓ 退出：按q键

文件显示命令

✓ head命令--显示文件的开头几行

✓ 格式

head [选项] 文件名 ...

✓ 选项

-num 显示前num行，默认值是10

-cn 显示前面n个字节

✓ 例子

head -5 file1

head -10c file1

head file1 file2

文件显示命令

✓ tail命令--显示文件的末尾几行

✓ 格式

tail [选项] 文件名 ...

✓ 选项

-num 显示最后num行，默认值是10

-cn 显示最后n个字节

✓ 例子

tail -20 file1 //显示文件的倒数20行内容

tail +20 file1 //显示从第20行到末尾的内容

文件显示命令

✓ touch命令--change file timestamps

✓ 功能

- ✓ 修改文件的时间为系统当前的时间

- ✓ 如果文件不存在，则创建一个空文件

✓ 格式

touch [格式] 文件名...

✓ 例子

touch abc

查找命令

✓ find命令--search for files in a directory hierarchy

✓ 功能

查找指定文件和目录

✓ 格式

find [搜索路径] [匹配标准]

✓ 选项

-name	根据文件名查找
-size	根据文件的大小查找
-user	根据文件的所有者查找
-ctime	根据修改时间查找
-type	根据文件类型查找

查找命令

✓ find命令

✓ 例子：根据名称查找

//查找系统中名为hosts的文件

```
find / -name hosts
```

//查找当前目录下h开头的文件

```
find . -name "h*"
```

//查找/etc目录下文件名包含h的文件

```
find /etc -name "*h*"
```

查找命令

✓ find命令

✓ 例子：根据大小查找

//找出/home大小为100个数据块的文件

`find /home -size 100b` (b可以省略)

//查找当前目录下小于100字节的文件

`find . -size -100c`

//查找当前目录下大于10K字节的文件

`find . -size +10k`

查找命令



✓ find命令

- ✓ 例子：根据文件的所有者查找

//找出/home下属于用户xx的所有文件

```
find /home -user xx
```

- ✓ 例子：根据文件的类型查找

//查找当前目录下的普通文件

```
find . -type f
```

查找命令



✓ find命令

✓ 例子：根据文件的时间查找

//查找当前目录下20天之内修改过的文件

```
find . -ctime -20
```

//查找当前目录下超过20天修改过的文件

```
find . -ctime +20
```

查找命令

✓ which命令--shows the full path of commands

✓ 功能

查找系统命令所在目录

✓ 格式

which [命令名]

✓ 例子

which ls

✓ 说明

显示命令的别名信息

查找命令

✓ grep命令--print lines matching a pattern

✓ 功能

在文件中搜索字符串匹配的行

✓ 格式

grep [选项] 匹配字符串 文件名

查找命令

✓ grep命令--print lines matching a pattern

✓ 例子

```
grep xx file1
```

```
grep -n xx file1
```

```
grep -c xx file1
```

✓ 选项

选项	说明
-v	列出不匹配串或正则表达式的行。
-c	对匹配的行计数。
-l	只显示包含匹配的文件的文件名。
-i	产生不区分大小写的匹配，缺省状态是区分大小写

排序命令



✓ sort命令

✓ 功能

将逐行对文件中的内容进行排序

✓ 格式

sort [选项] 文件

✓ 选项

-d 按字母顺序排序

-r 按逆序输出排序结果

复制、删除和移动文件的命令

✓ cp命令--copy files and directory

✓ 功能

复制文件或目录

✓ 格式

cp [选项] 源文件 目标文件/目录

✓ 选项

-f 覆盖已经存在的目标文件，不给提示

-i 覆盖已经存在的目标文件，给出提示

-r 复制目录，包含目录下的各级子目录

复制、删除和移动文件的命令

✓ cp命令--copy files and directory

✓ 说明

命令中的参数可以是相对路径，也可以是绝对路径
可以使用通配符

✓ 例子

//复制文件 file1，并命名为 file2

```
cp file1 file2
```

//将所有C语言程序拷贝至 dir1 目录(相对路径)中

```
cp *.c dir1
```

//复制目录

```
cp -r dir1 dir2
```

复制、删除和移动文件的命令

✓ `rm` 命令--remove files or directories

✓ 功能

删除文件或目录

✓ 格式

`rm [选项] 文件名...`

✓ 选项

`-i` 交互式的删除文件

`-f` 强制删除文件，没有提示

`-r` 递归的删除指定目录及其子目录和文件

复制、删除和移动文件的命令

✓ rm 命令

✓ 例子

//删除dir目录（含dir目录下所有文件和子目录）

```
rm -r dir
```

//强行删除dir目录、不加询问

```
rm -rf dir
```

复制、删除和移动文件的命令

✓ mv命令--move (rename) files

✓ 功能

移动文件/给文件改名

✓ 格式

mv [选项] 源文件 目标文件/目录

✓ 选项

-i 交互式操作（默认）

-f 禁止交互式操作

复制、删除和移动文件的命令

✓ mv命令--move (rename) files

✓ 例子

//将a.txt文件移动上层目录

```
mv a.txt ../
```

//将a.txt改名为b.txt

```
mv a.txt b.txt
```

//将dir2目录上移一层

```
mv dir2 ../
```

文件内容统计命令

✓ wc命令

✓ 功能

统计文件中的行数、字数、字节数

✓ 格式

wc [选项] 文件名

✓ 选项

-l 统计行数line

-w 统计字数（以字符串为单位）

-c 统计字节数bytes

目录的操作命令



- ✓ 改变工作目录
- ✓ 显示目录内容
- ✓ 创建和删除目录
- ✓ 链接文件
- ✓ 改变文件或目录存取权限
- ✓ 改变用户组和文件主

改变工作目录

✓ cd命令--change directory

✓ 格式

cd 目录名

✓ 例子

cd /usr/bin //切换到/usr/bin目录

cd dir1 //切换到当前目录下的dir1子目录

cd .. //切换到上一层目录

cd / //切换到系统根目录

cd //切换到用户主目录

显示当前路径

✓ pwd命令--print working directory

✓ 格式

pwd

显示目录内容



✓ ls命令--list directory contents

✓ 格式

ls [选项] 目录或文件

✓ 选项

-a 显示隐藏文件

-A 功能同上，不列出“.”、“..” 目录

-l 使用长格式显示文件内容

-d 显示目录属性

显示目录内容



✓ ls命令--list directory contents

✓ 选项

-R 显示目录及下级子目录结构

-F 在列出的文件名后面加上符号以区分文件的类型

/ 表示目录

* 表示可执行文件

@ 表示符号链接文件

显示目录内容

✓ ls命令--列出指定目录的内容

```
[root@pc root]# ls -l install.log
-rwxr--r-- 1 root root 26195 Dec 17 10:42 install.log
```

符号	文件类型
-	普通文件
d	目录
b	块设备文件
c	字符设备文件
l	符号链接文件

符号	存取权限
r	读 read
w	写 write
x	执行 execute
-	没有设置权限

创建和删除目录



✓ mkdir命令--make directories

✓ 格式

mkdir [选项] 目录名

✓ 选项

-m 数字 对新建目录设置存取权限

-p 一次建立多级目录

创建和删除目录



✓ mkdir命令--make directories

✓ 例子

`mkdir dir1`

//在当前目录下创建dir1目录

`mkdir dir1 dir2`

//建立2个目录

`mkdir /root/dir3`

//根据绝对路径创建目录

`mkdir -p bin/dir4`

//当bin不存在时，同时创建父目录和子目录

创建和删除目录

✓ rmdir命令--remove empty directory

✓ 格式

rmdir [选项] 目录名

✓ 说明

如果该目录中仍有其他文件，那就不能用rmdir命令把它删除。该目录中的文件必须用rm命令先删除

✓ 选项

-p 递归删除目录

当子目录删除后，如果父目录也为空时，也一同被删除

当子目录删除后，如果父目录不为空时，则保留父目录

创建和删除目录



✓ rmdir命令--remove empty directory

✓ 例子

```
rmdir dir1
```

```
rmdir -p bin/dir1
```

链接文件



✓ 功能

- ✓ 该命令在文件之间创建链接，这种操作实际上是给系统中已有的某个文件指定另外一个可用于访问它的名称

✓ 优点

- ✓ 对于这个新的文件名，可以为之指定不同的访问权限，以控制对信息的共享和安全性问题。

✓ 链接种类

- ✓ 硬链接
- ✓ 符号链接

链接文件

✓ 硬链接 (Hard Link)

✓ 格式

`ln 源文件 链接名`

✓ 例子

`ln file1 test1`

✓ 说明

文件共享，二者在物理上是同一文件。 `ls -li`

引用计数增加

链接文件



✓ 硬链接 (Hard Link)

✓ 说明

若链接名是目录名，将在该目录之下建立与目标文件同名的链接文件。

✓ 例子

In file1 dir1

链接文件



✓ 符号链接 (Symbolic Link)

✓ 格式

`ln -s 源文件 链接名`

✓ 例子

`ln -s /home/wuhua/test1 file1`

链接文件



✓ 符号链接 (Symbolic Link)

✓ 说明

建立符号链接时，源文件要使用绝对路径名

符号链接是一种特殊的文件类型

引用计数不变

链接文件



✓ 符号链接和硬链接的区别

- ✓ 符号链接是一种特殊的文件类型
- ✓ 在硬链接中，链接数会增加；在符号链接中，链接数为1
- ✓ 在硬链接中，删除源文件，文件不会被真正删除，只是链接数减1；在符号链接中，删除源文件，源文件将被删除，其他链接文件将无法继续访问

改变文件或目录存取权限

✓ chmod命令--change file access permissions

✓ 用于改变或设置文件/目录的存取权限

✓ 两种方法

符号模式：用字符串设置权限

绝对方式：使用八进制数设置权限

改变文件或目录存取权限

✓ 用字符串设置权限

✓ 格式

chmod key 文件名

key = [操作对象] [操作符号] [用户权限]

操作对象	操作符号	用户权限
u (user)主用户	+ 增加权限	r 读
g (group)组用户	- 取消权限	w 写
o (others)其他用户	= 重新赋予权限	x 执行
a (all)所有用户		

改变文件或目录存取权限

✓ 用字符串设置权限

✓ 格式

chmod key 文件名

key = [操作对象] [操作符号] [用户权限]

✓ 例子

chmod ugo+x testfile

chmod a+x testfile

chmod ug+w,o-w testfile

改变文件或目录存取权限

✓ 使用八进制数设置权限

✓ 格式

chmod mode 文件名

mode	<u>r</u> <u>w</u> <u>x</u>	<u>r</u> <u>-</u> <u>x</u>	<u>r</u> <u>-</u> <u>-</u>
二进制	<u>1</u> <u>1</u> <u>1</u>	<u>1</u> <u>0</u> <u>1</u>	<u>1</u> <u>0</u> <u>0</u>
八进制	7	5	4

✓ 例子

chmod 754 testfile

chmod 644 testfile

chmod 777 testfile

改变文件或目录存取权限

✓ umask命令

✓ 功能

用来设置文件的缺省权限的掩码

```
[root@localhost ch02]# ll file1
-rw-r--r--  1 root    root          15 Mar  4 17:59 file1
[root@localhost ch02]# ll -d dir1
drwxr-xr-x  2 root    root       4096 Mar  4 13:55 dir1
[root@localhost ch02]# _
```

改变文件或目录存取权限

✓ umask命令

✓ 掩码值的含义

掩码值	0	2	2
二进制	<u>0 0 0</u>	<u>0 1 0</u>	<u>0 1 0</u>
掩掉的权限	<u>- - -</u>	<u>- W -</u>	<u>- W -</u>

✓ 说明

缺省创建的文件没有可执行权限

如果想使文件有运行权限，只能由 `chmod` 命令进行设置

改变文件或目录存取权限

✓ umask命令

✓ 格式

umask mode

✓ 例子

umask

umask -S

umask 077

✓ 说明

运行umask命令只对当前shell环境起作用

改变文件或目录存取权限

✓ umask命令

✓ 说明

符号	存取权限	对文件	对目录
r	读 read	查看文件	查看目录的内容 ls
w	写 write	修改文件	可以在目录中创建、删除和移动 touch mkdir rm mv
x	执行 execute	执行文件	可以进入目录 cd

改变用户组和文件主

✓ useradd命令--create a new user

✓ 格式

useradd [选项] 用户名

✓ 例子

useradd user1

✓ 说明

只能由root用户使用

新建的用户暂时无法登陆，需要用passwd命令为其设置口令

改变用户组和文件主

✓ userdel命令--delete a user

✓ 格式

userdel [选项] 用户账号

✓ 选项

-r 用来删除用户主目录以及目录中文件

✓ 例子

userdel -r user1

改变用户组和文件主

✓ groupadd命令--增加用户组

✓ 格式

groupadd [选项] 用户组名

✓ 例子

groupadd group1

✓ 说明

ID值0~499是保留给系统账号使用

改变用户组和文件主

✓ groupdel--删除指定的组账号

✓ 格式

groupdel [群组名称]

✓ 例子

groupdel group1

✓ 说明

若该群组中仍包括某些用户，则必须先删除这些用户后，方能删除群组

改变用户组和文件主



✓ chown命令--change file owner and group

✓ 功能

改变某个文件或目录的所有者

✓ 格式

chown [选项] 用户 文件名

✓ 选项

-R 递归式的改变指定目录及其所有子目录、文件的文件主

改变用户组和文件主

✓ chown命令--change file owner and group

✓ 例子

`chown user1 testfile`

`chown user1 testdir`

✓ 说明

只有超级用户和文件主才能使用该命令

改变用户组和文件主

✓ chgrp命令--change group ownership

✓ 功能

改变文件或目录所属的用户组

✓ 格式

chgrp [选项] 组名 文件名

✓ 选项

-R 递归式的改变指定目录及其下面的所有子目录和文件的用户组

✓ 例子

chgrp group1 testfile

chgrp -R group1 testdir

查看联机帮助

✓ man命令--display the on-line manual pages

✓ 功能

显示某一命令的联机帮助手册

✓ 格式

man [选项] 命令名

✓ 选项

-f 只显示命令的功能

-w 只显示帮助文件所在的位置

--path 同上

查看联机帮助

✓ help命令

✓ 功能

查看shell内置命令的帮助信息。如：cd、echo、help、pwd、umask等

✓ 格式

help 命令

✓ 选项

-s 显示命令的格式

✓ 例子

help cd

help -s cd

进程管理



✓ w命令--查看系统中登陆用户及其进程信息

✓ 格式

w [选项]

✓ 例子

w

w wuhua

进程管理



✓ w命令

✓ 说明

load average: 分别显示系统在过去1、5、15分钟内的平均负载

IDLE: 用户闲置时间

JCPU: 终端上所有进程消耗的CPU时间

PCPU: 当前执行程序耗费时间

WHAT: 当前执行命令

进程管理



✓ ps命令

✓ 功能

查看系统中正在运行的进程

✓ 格式

ps [选项]

进程管理

✓ ps命令

✓ 例子：显示当前shell启动的所有进程

```
[wuhua@localhost ~]$ ps
  PID TTY          TIME CMD
 58021 pts/0        00:00:00 bash
 58066 pts/0        00:00:00 ps
```

PID: 进程编号

TTY: 进程所运行的位置（终端）

TIME: 进程占用的CPU处理时间

CMD: 进程所运行的命令

进程管理

✓ ps命令

✓ 例子：查看指定用户进程

```
[wuhua@localhost ~]$ ps -u wuhua
```

PID	TTY	TIME	CMD
1762	?	00:00:00	gnome-keyring-d
1781	?	00:00:00	gnome-session-b
1788	?	00:00:00	dbus-launch
1789	?	00:00:00	dbus-daemon

进程管理

✓ ps命令

✓ 例子：列出所有的正在运行的进程

```
[wuhua@localhost ~]$ ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.4	193700	4392	?	Ss	01:31	0:05	/usr/lib/systemd/systemd
root	2	0.0	0.0	0	0	?	S	01:31	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	01:31	0:01	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	01:31	0:00	[kworker/0:0H]

%CPU 进程的cpu占用率

%MEM 进程的内存占用率

VSZ 进程所使用的虚存的大小 (Virtual Size)

RSS 进程使用的实际内存的大小

进程管理

✓ ps命令

✓ 例子：列出所有的正在运行的进程

```
[wuhua@localhost ~]$ ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.4	193700	4392	?	Ss	01:31	0:05	/usr/lib/systemd/systemd
root	2	0.0	0.0	0	0	?	S	01:31	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	01:31	0:01	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	01:31	0:00	[kworker/0:0H]

✓ STAT: 进程的状态

R 正在执行

S 睡眠状态

s 包含子进程

< 高优先级

进程管理

✓ ps命令

✓ 例子：显示进程的树形结构

```
[wuhua@localhost ~]$ pstree
systemd—ModemManager—2*[{ModemManager}]
        |
        |—NetworkManager—dhclient
        |                  |
        |                  |—2*[{NetworkManager}]
        |
        |—VGAuthService
        |—abrt-watch-log
        |—abrt-d
        |—accounts-daemon—2*[{accounts-daemon}]
        |—agetty
        |—alsactl
```

进程管理



✓ top命令

✓ 功能

实时监控进程状况， top屏幕自动每5秒刷新一次

进程管理

✓ top命令

系统状态

```
top - 21:18:55 up 19:47, 3 users, load average: 0.00, 0.01, 0.05
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.0 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 999696 total, 89716 free, 535156 used, 374824 buff/cache
KiB Swap: 2097148 total, 1774784 free, 322364 used. 229184 avail Mem
```

进程信息

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
284	root	20	0	0	0	0	S	0.3	0.0	0:03.69	xfsaild/sda2
58800	wuhua	20	0	157716	2236	1548	R	0.3	0.2	0:00.05	top
1	root	20	0	193700	4392	2520	S	0.0	0.4	0:05.37	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.03	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:01.20	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	0:01.27	rcu_sched
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.39	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
13	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
14	root	20	0	0	0	0	S	0.0	0.0	0:00.03	khungtaskd
15	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd

进程管理



✓ top命令

✓ top命令中的交互命令

空格 立即刷新

h 进入帮助

q 退出程序

P 按CPU使用率排序

M 按使用内存大小排序

u 查看指定用户的进程

进程管理

✓ sleep命令

✓ 功能

使进程暂停执行一段时间

✓ 格式

sleep time

✓ 说明

time为进程将睡眠的时间，以秒为单位

✓ 例子

sleep 10; who

//10秒钟后执行命令who

进程管理



✓ kill命令--terminate a process

✓ 功能

向一个或多个进程发送终止信号

✓ 格式

kill [-信号] 进程号

✓ 例子

kill -l

kill 2336

kill -9 2336

进程管理

✓ kill命令

✓ 例子

```
ps -ef|grep vim
```

```
kill xxx
```

压缩和解压缩



- ✓ 压缩gzip和解压缩gunzip
- ✓ 压缩zip和解压缩unzip
- ✓ 压缩bzip2和解压缩bunzip2
- ✓ 文件打包tar

压缩和解压缩

✓ tar命令--对文件和目录进行打包

✓ 格式

tar [选项] 打包文件 文件/目录

✓ 选项

-c

产生.tar打包文件

-v

显示详细信息

-f

指定压缩后的文件名

-z

利用gzip进行压缩/解压缩

-j

利用bzip2进行压缩/解压缩

-x

解包

压缩和解压缩



✓ tar命令--对文件和目录进行打包

✓ 例子：打包并压缩

//将目录dir打包并压缩成一个文件，后缀为 .tar.gz

```
tar -zcvf dir.tar.gz dir
```

//解包解压缩到当前目录下

```
tar -zxvf dir.tar.gz
```

//解包解压缩到指定目录dir2下

```
tar -zxvf dir.tar.gz -C dir2
```