

Linux 101 进程 服务 任务*

- 进程是什么
- 作业控制
- 进程的监控与管理
- 服务是什么
- 服务的监控与管理
- 定时任务（一次性、周期性）

*以 Ubuntu 18.04 为例

走近进程 (1)

Shell 命令 *

桌面环境

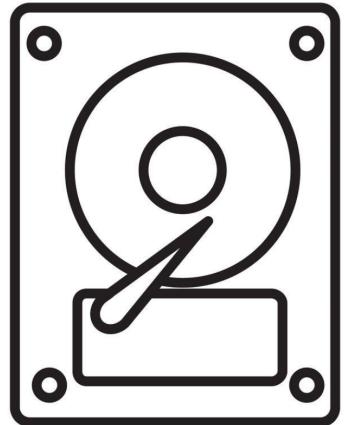
办公套件

游戏

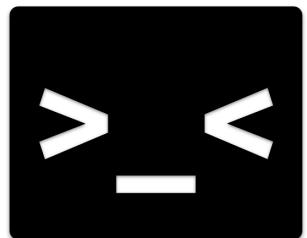
聊天软件

系统服务

走近进程 (2)

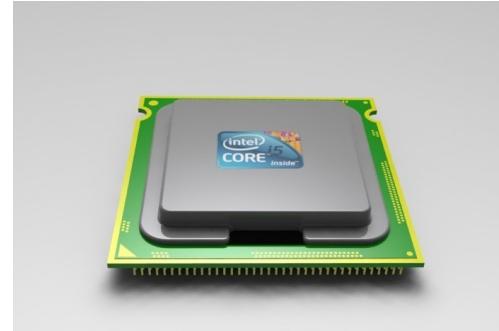


硬盘

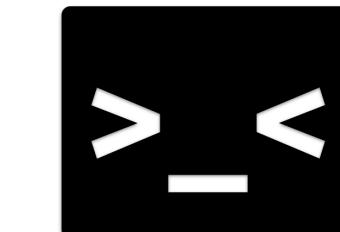
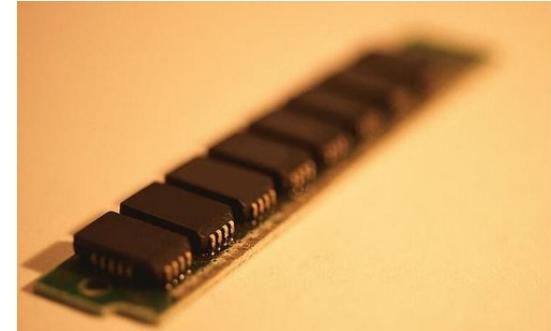


Shell (程序)

CPU

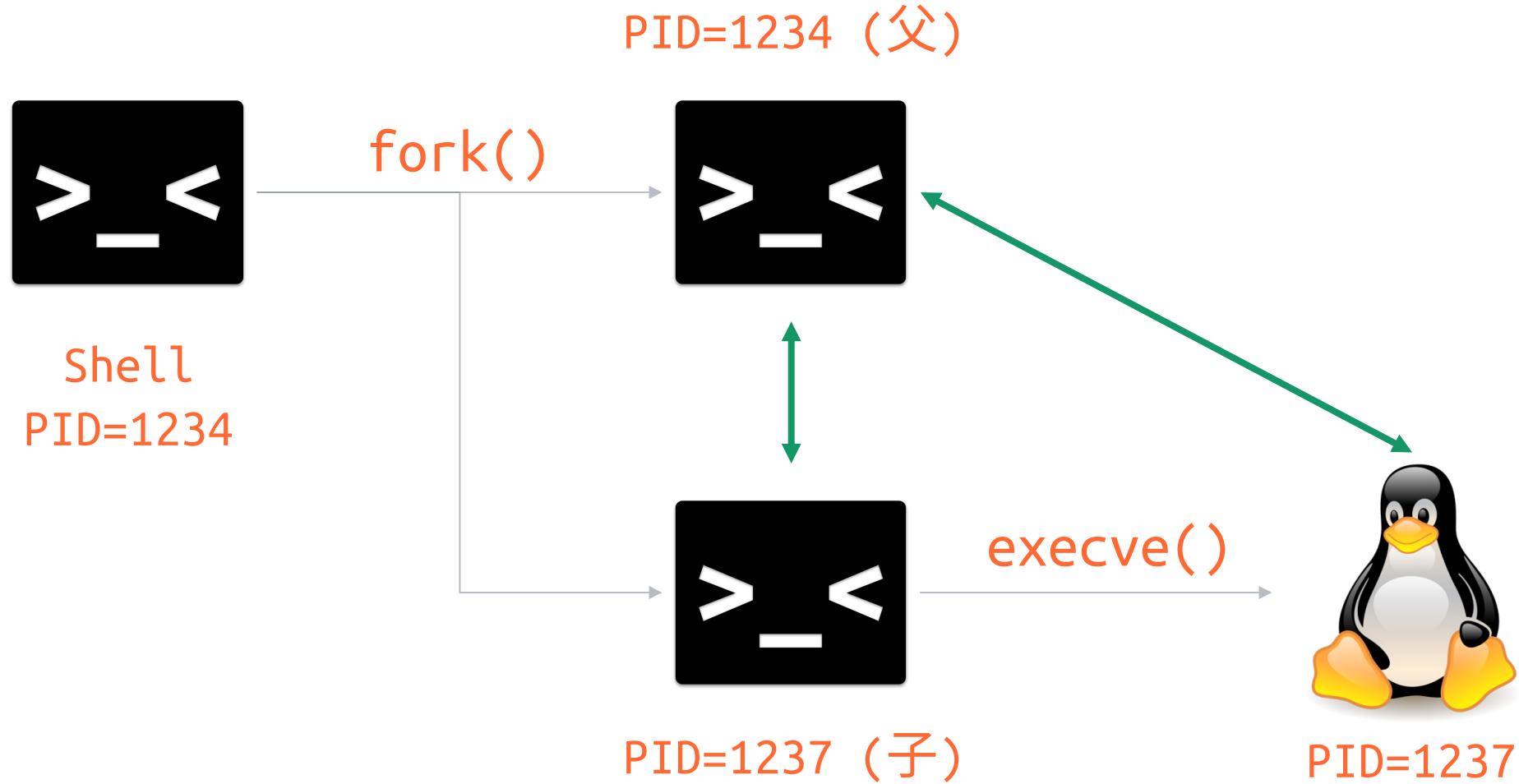


内存

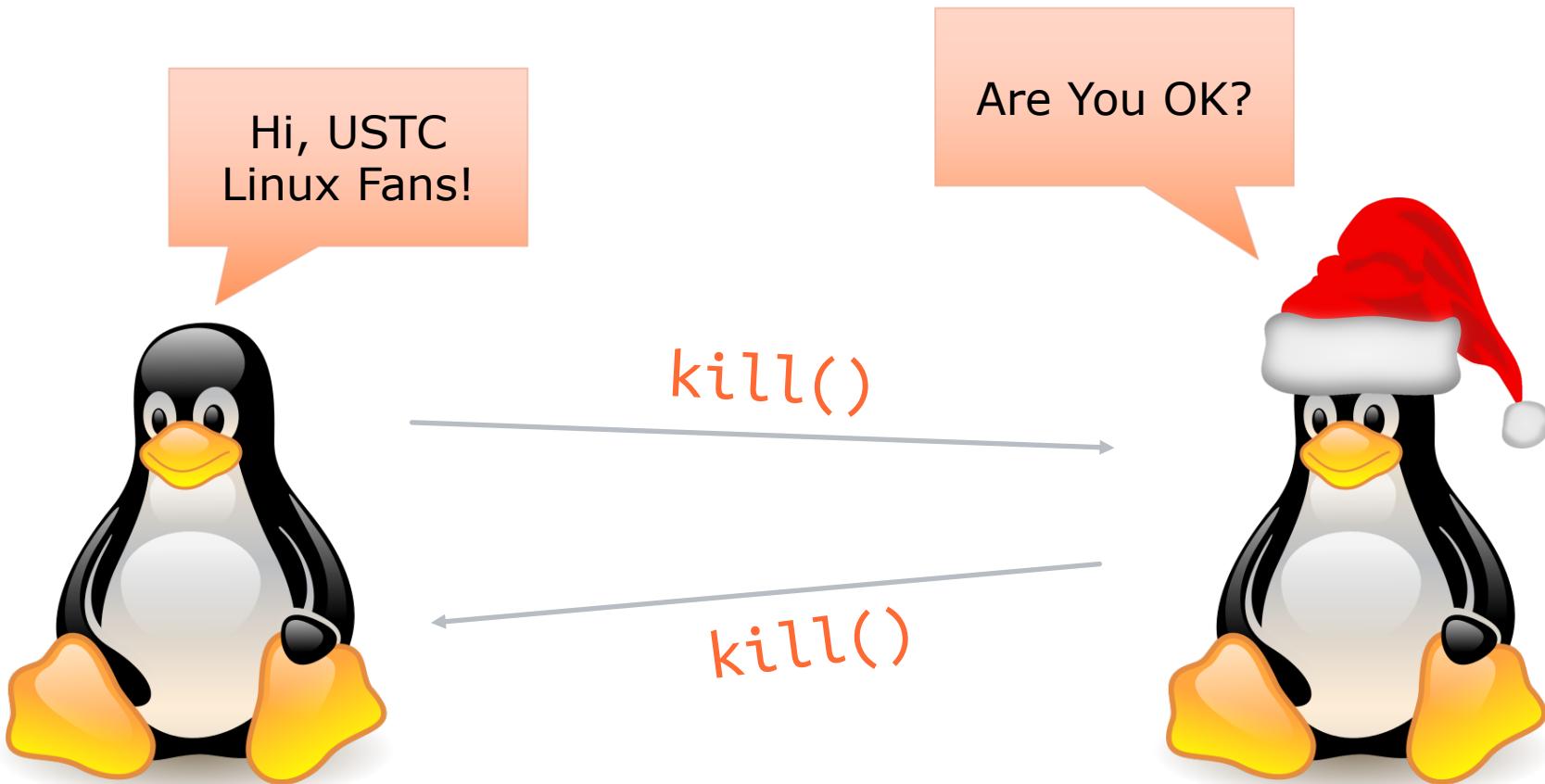


Shell (进程)

走近进程 (3)



信号 (1)



信号 (2)

\$ man 7 signal

常用信号	意义	默认行为	产生手段
SIGINT (interrupt)	朋友，别干了	终止进程	Ctrl-C
SIGTERM (terminate)	请优雅地死去	终止进程	kill <PID> pkill <进程名>
SIGKILL (kill)	请立即去世	终止进程	kill -9 <PID> pkill -9 <进程名>
SIGSEGV (segment violation)	这个进程在试图做一些不该做的事情	核心转储	(你的/别人的) 程序写得太烂了
SIGSTOP (stop) SIGTSTP (stop)	让某个进程变成植物人	停止进程	Ctrl-Z
SIGCONT (continue)	让植物人苏醒	继续进程	fg bg

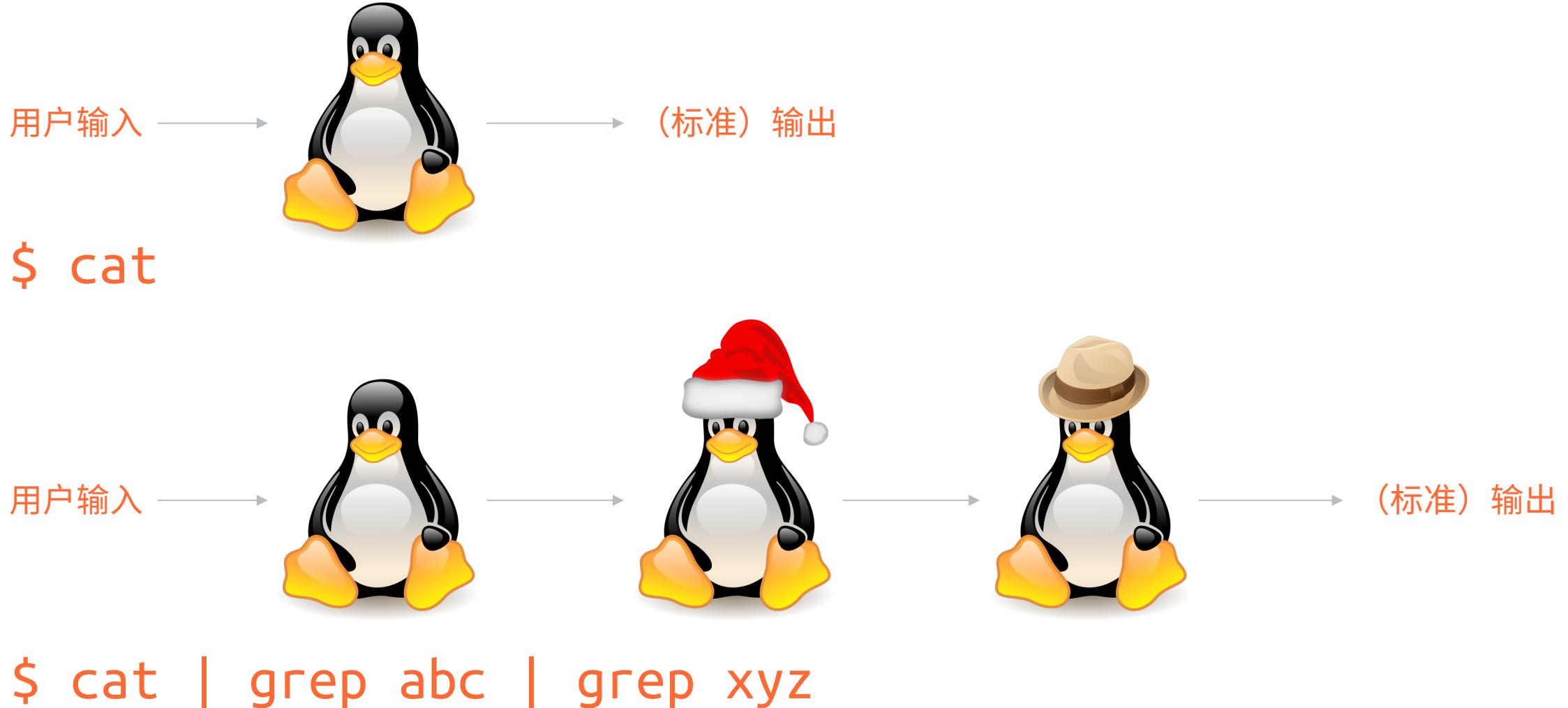
信号 (3) 发送信号之 kill 和 pkill

user@hostname:~\$ kill -<信号> <PID>

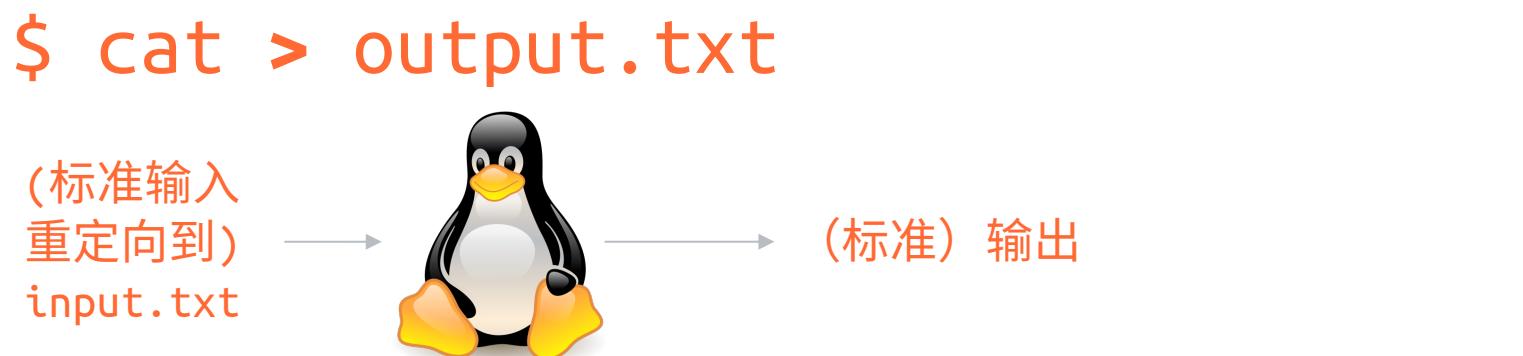
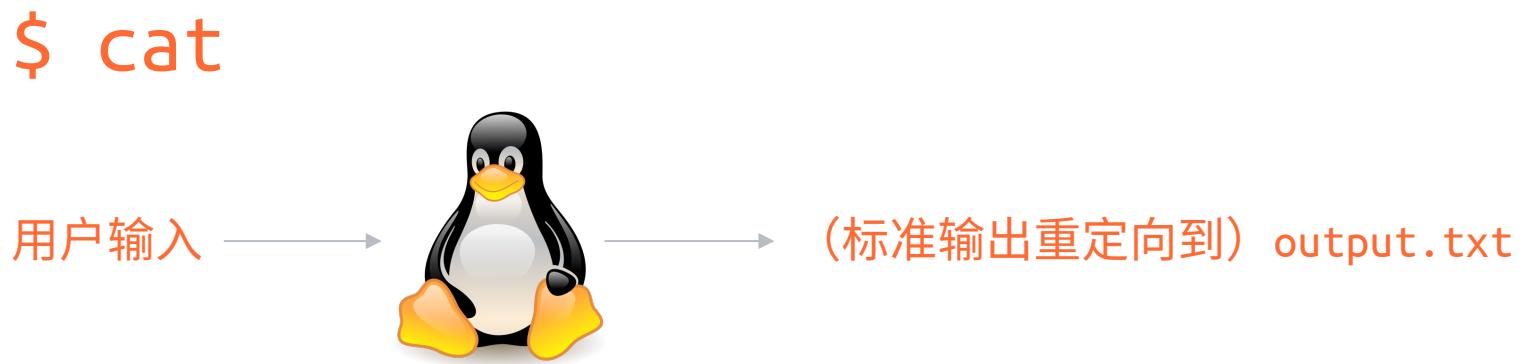
user@hostname:~\$ pkill -<信号> <进程名>

- 信号可以使用数字和名称两种形式表示
 - kill \$PID: 向 PID 发送 SIGTERM 信号
 - kill -9 \$PID: 向 PID 发送 SIGKILL
 - pkill -HUP rsyslogd: 向 rsyslogd 发送 SIGHUP 信号
- pkill 默认会向所有名字匹配的进程发送信号
- pgrep 找出进程的 PID 但不发送信号

作业简介 (1) 管道



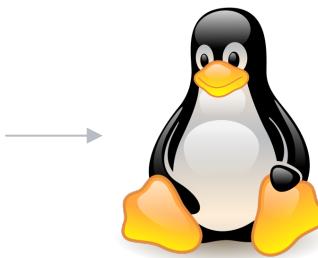
作业简介 (2) 文件重定向



\$ cat < input.txt

作业简介 (3) 文件重定向

(标准输入
重定向到)
`input.txt`



(标准输出重定向到) `output.txt`

```
$ cat < input.txt > output.txt
```

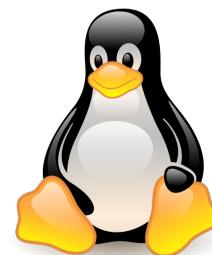
用户输入



(标准错误重定向到) `err_output.txt`

```
$ cat 2> err_output.txt
```

用户输入



所有输出（标准输出 & 错误）都扔到黑洞（`/dev/null`）里去！

```
$ cat > /dev/null 2>&1
```

Wait a minute...

- 连上实验室 SSH，貌似只有一个命令行界面……
- 如果这个作业很花时间，那在这期间我岂不是什么都做不了？
- 如何让多个作业同时运行？
- 如何暂停一个作业，然后恢复运行？



作业控制 (1) 前台作业与后台作业

```
user@hostname:~$ ping www.ustc.edu.cn
```

```
PING www.ustc.edu.cn (218.22.21.21) 56(84) bytes  
of data.
```

(下略)

静静地看着输出，等待着结束……

作业编号为 1

管线中最后一个进程的
PID 为 123

表示该作业
在后台执行

```
user@hostname:~$ ping www.ustc.edu.cn &  
[1] 123
```

好了，可以继续工作了……

```
user@hostname:~$ PING www.ustc.edu.cn  
(218.22.21.21) 56(84) bytes of data.  
(下略)
```

怎么一直在输出？好像没法终止？按 Ctrl+C 也没用！

作业控制 (2) 暂停前台作业

```
user@hostname:~$ ping www.ustc.edu.cn
```

```
PING www.ustc.edu.cn (218.22.21.21) 56(84) bytes of data.
```

```
64 bytes from 202.38.64.246: icmp_seq=1 ttl=62 time=1220 ms
```

^Z

```
[2]+ Stopped
```

```
user@hostname:~$
```

```
ping www.ustc.edu.cn
```

- 用 Ctrl-Z 停止一个前台作业。(SIGTSTP)

- 暂停的作业是后台作业吗?

- 暂停的进程被终止了吗?

- 能不能用 Ctrl-Z 停止后台作业?

作业控制 (3) 列出当前作业 jobs

```
user@hostname:~$ jobs
```

```
[1]- Stopped
```

```
[2]+ Stopped
```

```
user@hostname:~$
```

作业编号

作业状态

```
ping www.ustc.edu.cn
```

```
ping www.ustc.edu.cn
```

作业的完整命令

- 用 `jobs` 列出当前作业。
- `+` 表示 “当前作业” (如果有 `Stopped`, 则是最近停止的作业; 如果全部作业都在运行, 则是最近启动的作业)。
- `-` 表示当前作业退出后会成为当前作业的作业。

作业控制 (3) 列出当前作业 jobs

```
user@hostname:~$ jobs -l
```

```
[1]- 22563 Stopped      ping www.ustc.edu.cn  
[2]+ 22645 Stopped      ping www.ustc.edu.cn
```

```
user@hostname:~$ jobs -p
```

```
22563
```

```
22645
```

~~\$ man jobs~~

\$ man 1 bash

\$ man 7 builtins

作业控制 (4) 在后台恢复作业执行 bg

```
user@hostname:~$ jobs
```

```
[1]- Stopped
```

```
ping www.ustc.edu.cn
```

```
[2]+ Stopped
```

```
ping www.ustc.edu.cn
```

```
user@hostname:~$
```

```
bg %1
```

在后台恢复执行作业 1

```
user@hostname:~$
```

```
bg
```

在后台恢复执行当前作业
(“当前作业”可以用 % 或 %+ 表示)

- bg <jobspec> 在后台恢复执行作业
- %- 表示什么作业？
- 可以使用 bg 将前台任务放到后台吗？

作业控制 (5) 在前台执行作业 fg

```
user@hostname:~$ jobs
```

```
[1]- Stopped
```

```
[2]+ Stopped
```

```
user@hostname:~$
```

```
fg sleep
```

```
user@hostname:~$
```

```
fg ping
```

```
sleep 10
```

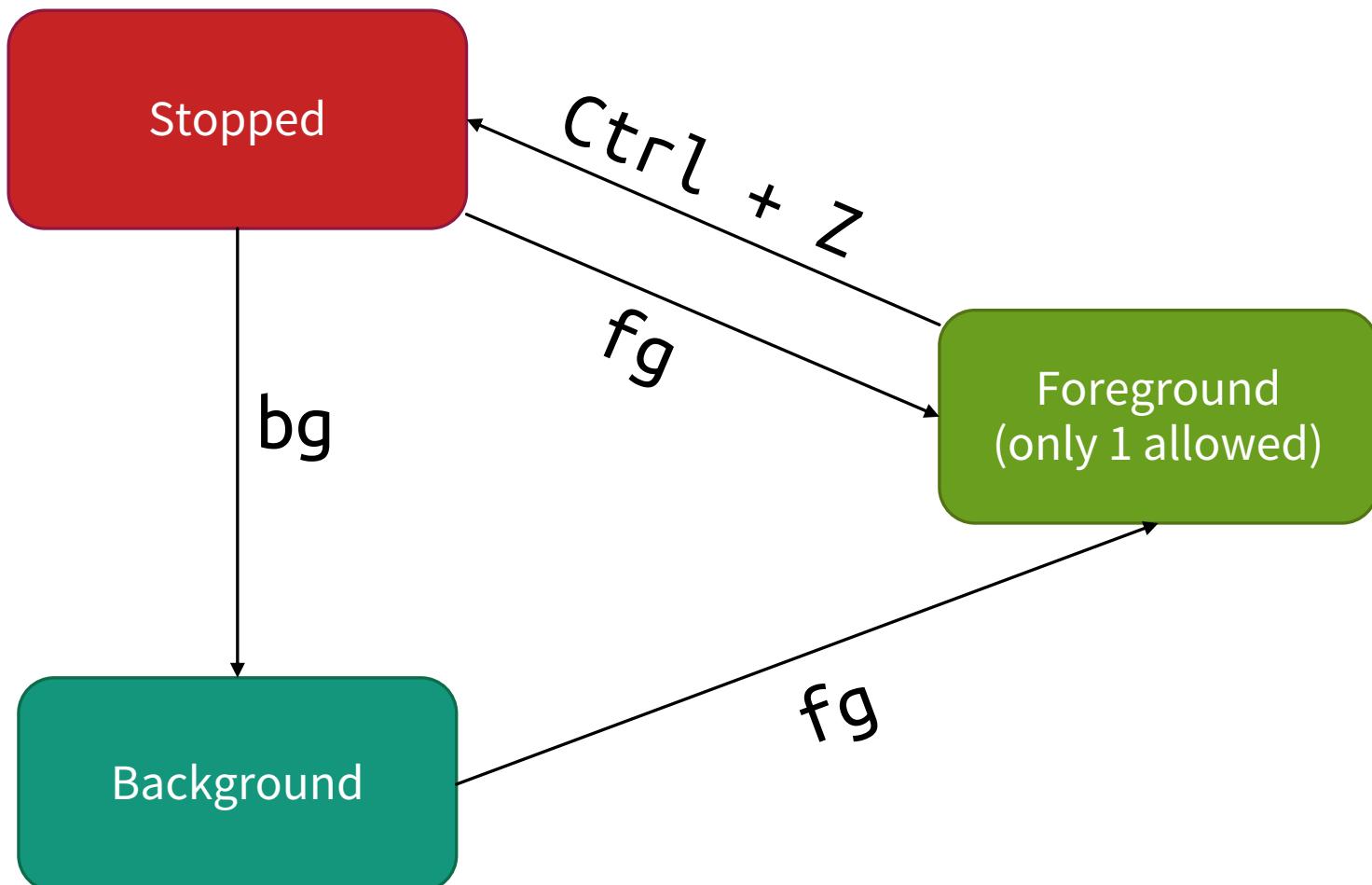
```
ping www.ustc.edu.cn
```

恢复执行 sleep 命令

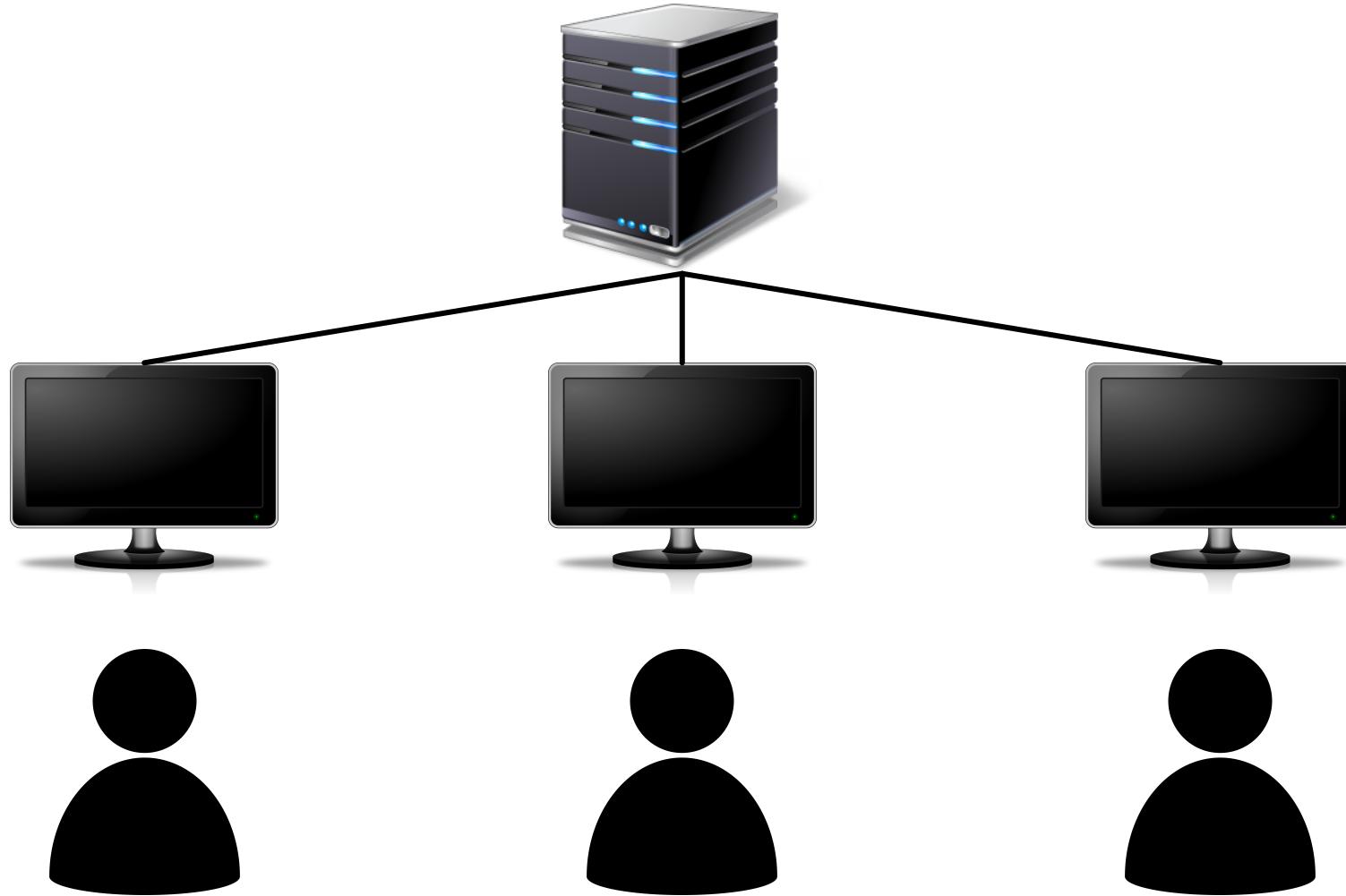
恢复执行 ping 命令

- fg <jobspec> 在前台执行作业
- bg 也可以用命令名
- fg 可以无参数，也可以用 %<编号> 和 %% 和 %+ 和 %- 作为参数
- 如果有两个命令同名的作业，这样做会发生什么？
- 可以用 fg 将后台任务放到前台吗？
- 试查找 jobspec (job specification) 语法的资料

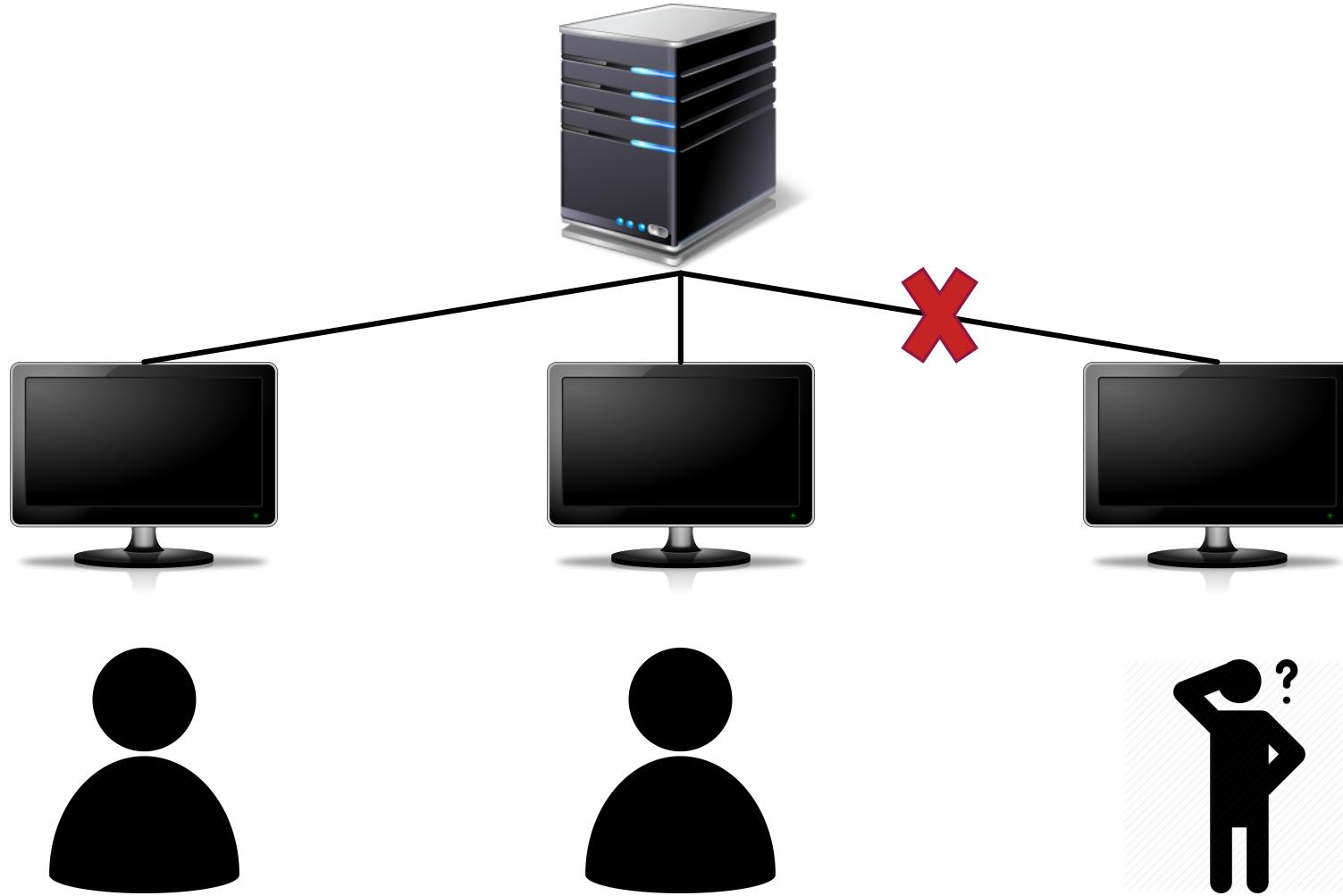
作业控制 (6) 作业状态图



一点历史 (1)

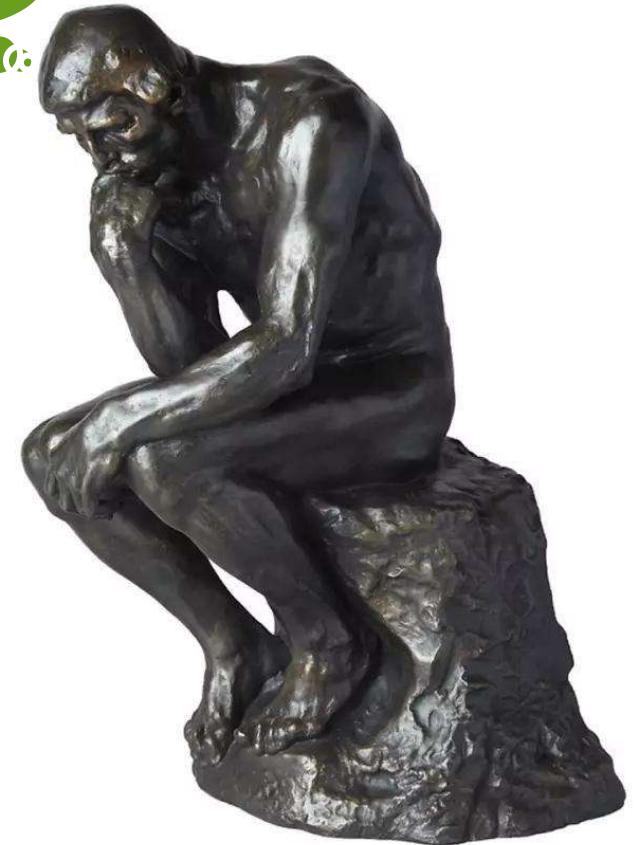


一点历史 (2)



Wait a minute...

- 如果我确实想让程序一直运行怎么做？
- 这种模型和“我通过 SSH 连接到实验室机器”似乎没有什么区别
- 那么一旦我断开 SSH 连接，程序就退出了，怎么办？



脱离控制 nohup

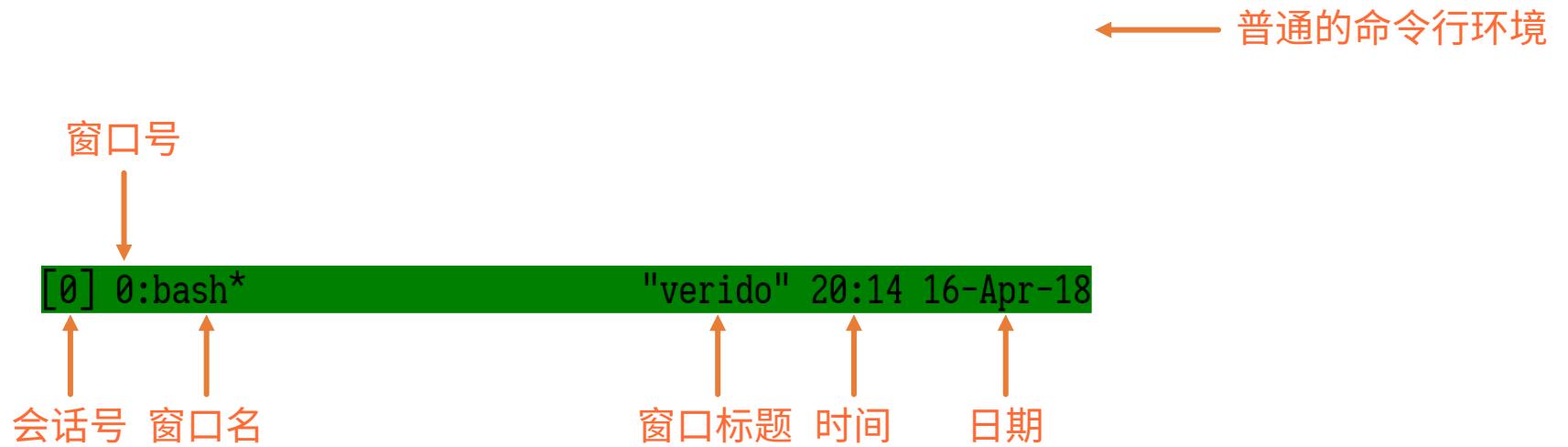
```
user@hostname:~$ nohup ping www.ustc.edu.cn
nohup: ignoring input and appending output to 'nohup.out'
^C
```

```
user@hostname:~$ cat nohup.out
PING www.ustc.edu.cn (202.38.64.246) 56(84) bytes of data.
64 bytes from 202.38.64.246: icmp_seq=1 ttl=62 time=3.94 ms
...(省略)...
--- www.ustc.edu.cn ping statistics ---
8 packets transmitted, 7 received, 12% packet loss, time 8871ms
rtt min/avg/max/mdev = 3.942/17.762/30.678/8.823 ms
```

- 忽略 SIGHUP 信号
- 如果 stdin 是终端 -> 重定向到一个不可读的文件
- 如果 stdout 是终端 -> 追加输出到 nohup.out

tmux (1)

ksqsf@verido:~\$



默认快捷键 **Ctrl + B** (C-b)
可以在 `~/.tmux.conf` 中修改

tmux (2)

普通的命令行环境 1 号

ksqsf@verido:~\$

普通的命令行环境 2 号

ksqsf@verido:~\$ █

[0] 0:bash*

"verido" 20:21 16-Apr-18

在右边新建窗口 C-b %

tmux (3)

普通的命令行环境 1 号

```
ksqsf@verido:~$ echo First!  
First!  
ksqsf@verido:~$
```

```
ksqsf@verido:~$ echo Second!  
Second!  
ksqsf@verido:~$ ■
```

[0] 0:bash*

"verido" 20:25 16-Apr-18

在下边新建窗口 C-b "

普通的命令行环境 2 号

tmux (4)

- 进程和终端脱节
- 完全使用 tmux session / window 机制管理 shell
- 可以把一个终端上的会话传递到另一个终端 (detach / attach)
- 超高可定制性
- 更多功能，有待你的发掘

↓
C-b d tmux attach
↓

快捷键	功能
C-b ?	列出所有快捷键
C-b C-z	暂停当前 tmux
C-b &	杀死当前窗口
C-b 0~9	选择 0~9 窗口
C-b \$	重命名当前会话
C-b ,	重命名当前窗口

查询进程状态 ps (1)

```
user@hostname:~$ ps
```

PID	TTY	TIME	CMD
2	tty1	00:00:00	bash
125	tty1	00:00:00	ping
126	tty1	00:00:00	ping
137	tty1	00:00:00	ping
546	tty1	00:00:00	ps

进程
ID

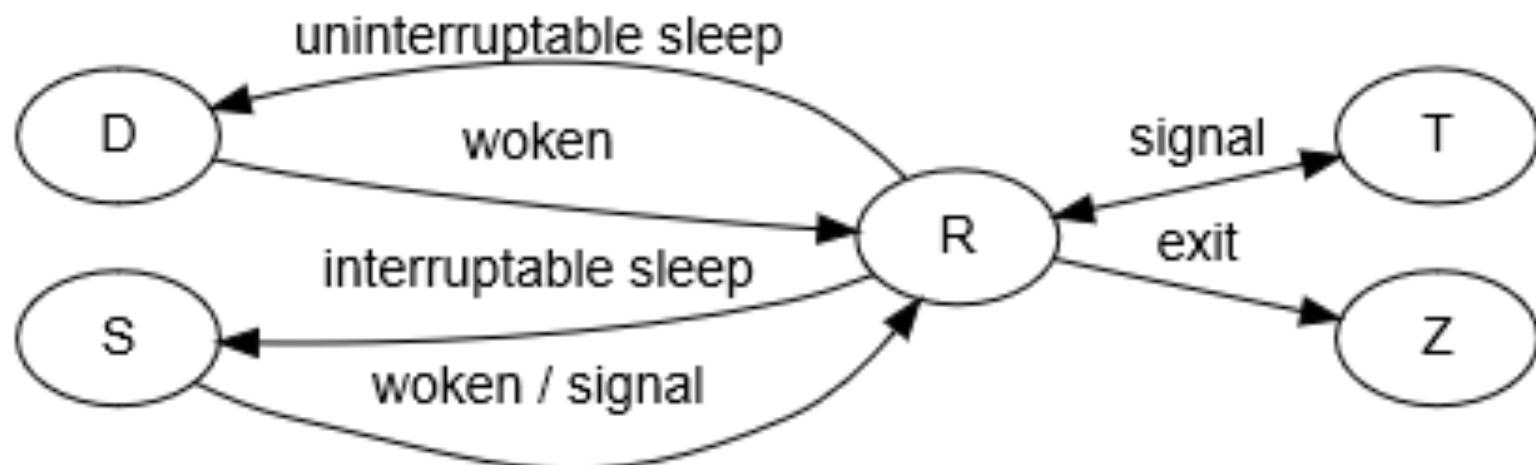
与进程关联
的终端

CPU 时间

进程命令

查询进程状态 ps (2)

状态	ps 表示	意义
Running	R	被调度运行
Interruptible	S	可中断的睡眠
Uninterruptible	D	不可中断的睡眠
Traced / Stopped	T	被追踪 / 暂停
Zombie	Z	僵死



查询进程状态 ps (3)

常见使用示例

```
user@hostname:~$ ps aux
```

```
user@hostname:~$ ps -ef
```

```
user@hostname:~$ ps -U root
```

```
user@hostname:~$ ps -U root -0 %mem --sort +%mem
```

```
user@hostname:~$ ps -C SomeMatrixCalc -0 %cpu
```

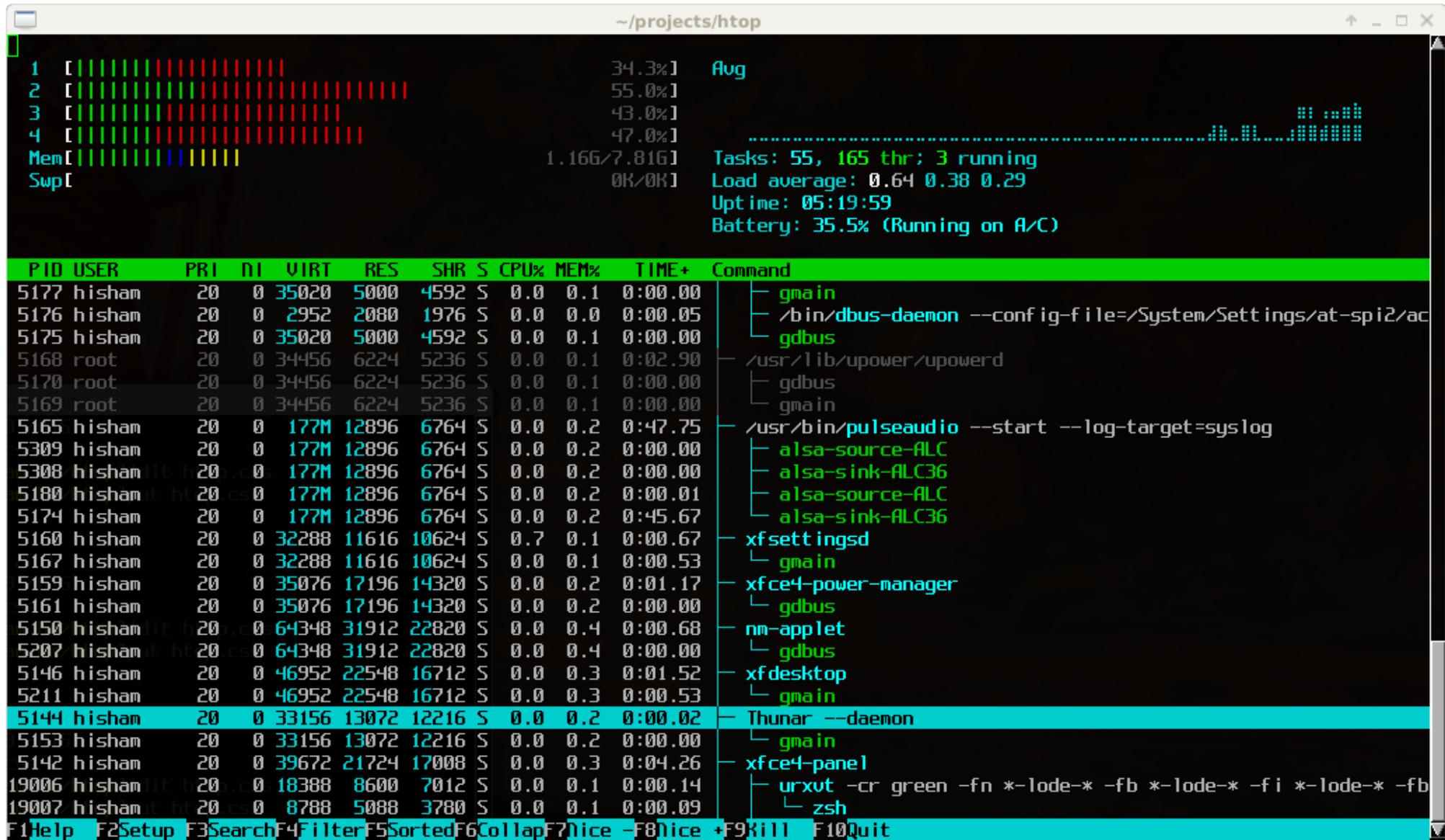
查看系统的内存占用 (1) free

	总量	未使用	tmpfs 使用	Buffer: 内核缓冲 Cache: 页缓存和 slab
内存 → Mem:	total 8311900	used 4262036	free 3813388	shared 17720
交换空间 → Swap:	25165824	45556	25120268	buff/cache 236476
已使用 (total - free - buffer - cache)				available 3909008

不交换的情况下新程序可用多少内存
(估计)

- 默认 KiB: -k / --kilo
- 换成其他单位: -b / --bytes, -m / --mega, -g / --giga, --tera
- 自动单位: -h / --human

查看进程的资源占用 (2) htop



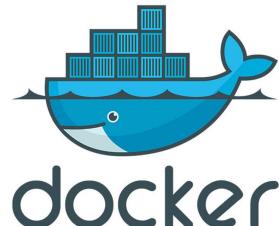
守护进程 (Daemon)



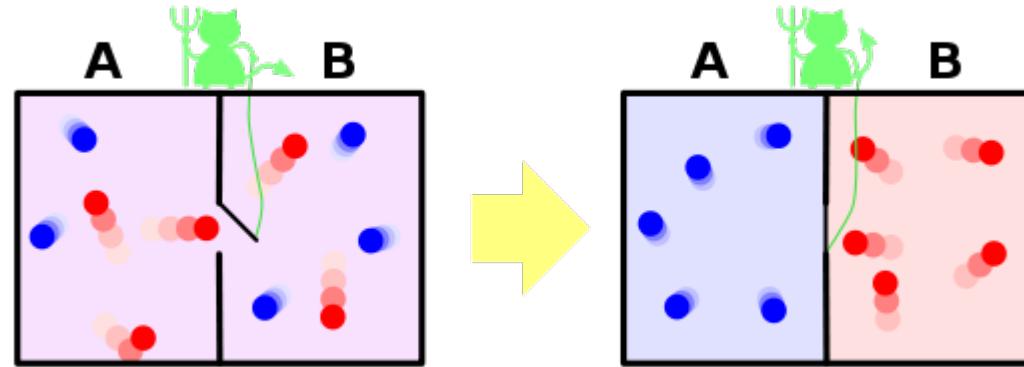
Apache



**systemd logind
crond sshd acpid**

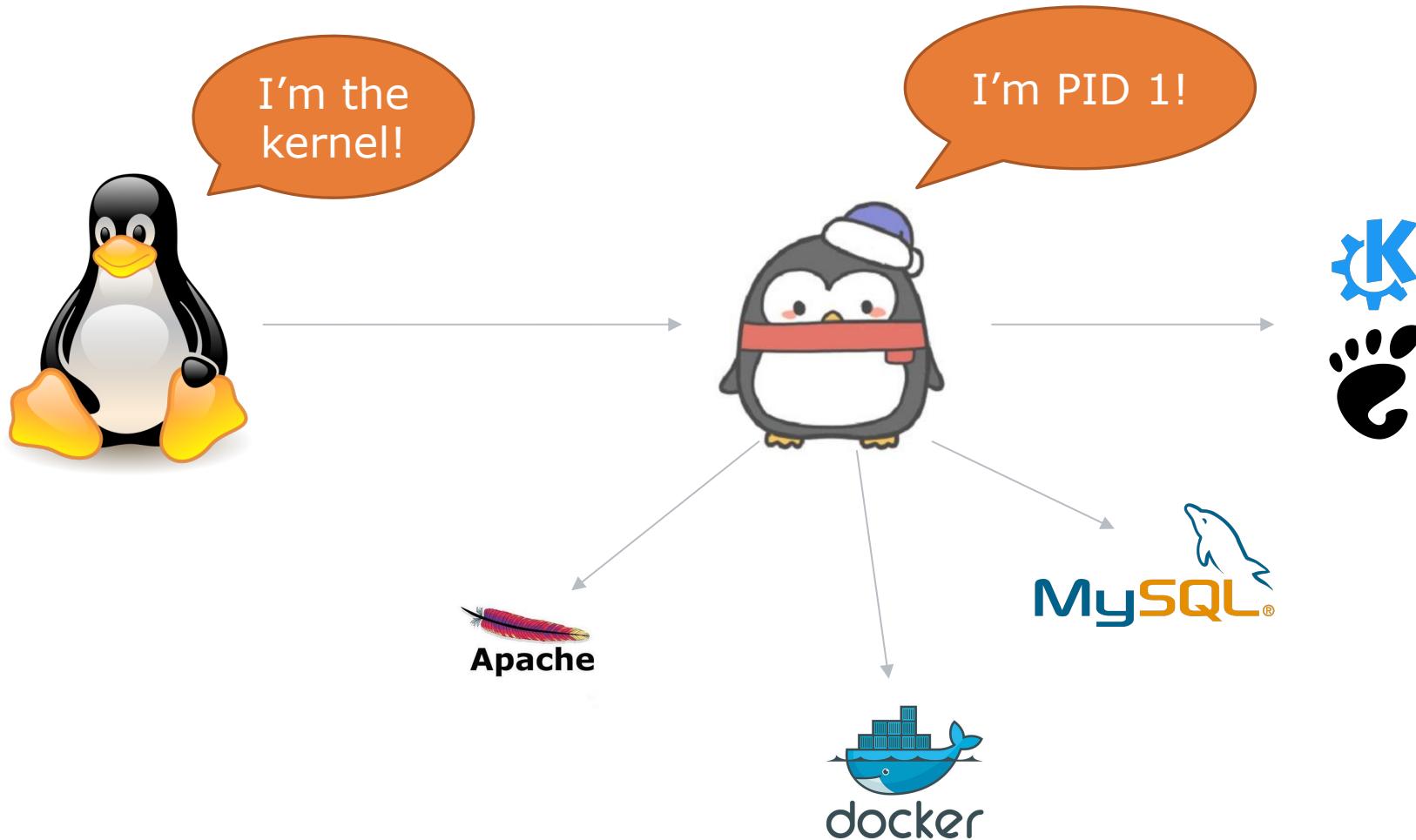


Daemon
(or, system services)



Maxwell's demon

服务管理 (1) init vs systemd



服务管理 (2) 单服务常见操作

root@hostname:~# systemctl stop apache2 停止 apache2

root@hostname:~# systemctl start nginx 启动 nginx

root@hostname:~# systemctl restart ssh 重新启动 ssh

root@hostname:~# systemctl reload nginx 让 nginx 重新加载配置文件

root@hostname:~# systemctl status httpd 显示 httpd 状态

root@hostname:~# systemctl enable nginx 开机自启动 nginx

root@hostname:~# systemctl disable apache2 取消开机自启动 httpd

服务管理 (3) 列出单元、查看日志

```
user@hostname:~$ systemctl list-units
root@hostname:~# journalctl [-f/- --follow]
[-x/- --catalog]
[-e/- --pager-end]
[-u/- --unit]=单元名
[-S/- --since]=起始时间
[-U/- --until]=截止时间
[ - --no-pager]
```

Wait a minute...

- 服务必须由我手动启用、禁用、启动、停止……
- 如果我想要监视计算的进度，能否定时周期性地通知我？
- 定时备份？
- 5 小时后自动发邮件？
- 有没有一种办法，能定时执行任务？



一次性任务 at (1)

```
user@hostname:~$ at 12:30          在 12:30 执行任务  
warning: commands will be executued using /bin/bash  
at> echo "hello" > /home/user/message  
at> <EOT>                      这里按下 Ctrl + D 结束输入  
job <N> at <DATETIME>        不要真的去敲 <EOT> 这 5 个字符啊 (
```



```
user@hostname:~$ at 17:23 2019-5-18 -f script.sh      在指定时间执行脚本 script.sh
```



```
user@hostname:~$ atq            列出当前用户的任务 (若为 root, 列出所有用户的任务)  
user@hostname:~$ at -c <N>      输出任务 N 脚本  
user@hostname:~$ atrm <N>       删除任务 N
```

*如果测试时候发现 at 没有在指定的时间执行命令，可能是时区设置有问题。
使用 sudo dpkg-reconfigure tzdata 修改系统时区设置。

一次性任务 at (2)

允许哪些用户使用 at?

/etc/at.allow

如果存在该文件，只有这些用户可以用 at

/etc/at.deny

如果存在该文件，则除此以外的用户都可以用 at

(root 用户不受任何约束)

如果这两个文件都存在呢?

如何让所有用户使用 at?

```
$ man 5 at.allow
```

周期性任务 (1) cron

/etc/crontab
crontab -e

系统 crontab
创建/修改本用户的 crontab

	分钟 (m)	小时 (h)	Day of month (dom)	月份 (month)	Day of week (dow)	用户 (user)	命令 (command)
示例	35	15	1,3-7	JAN-FEB	*	root	rm -rf /var/MyData
取值范围	0-59	0-23	1-31	1-12, JAN-DEC	0-6, SUN-SAT	(any user)	(any shell command)

@yearly
@monthly
@daily

@reboot
@weekly
@hourly

/etc/cron.hourly
/etc/cron.daily
/etc/cron.weekly
/etc/cron.monthly

周期性任务 (2) cron

	分钟 (m)	小时 (h)	Day of month (dom)	月份 (month)	Day of week (dow)	用户 (user)	命令 (command)
示例	35	15	1,3-7	JAN-FEB	*	root	rm -rf /var/MyData
取值范围	0-59	0-23	1-31	1-12, JAN-DEC	0-6, SUN-SAT	(any user)	(any shell command)

- 每周四 13:00 提醒我去补大物实验预习报告?
- 每周日 1:00 备份一下网站数据库?
- 每过一小时提醒我休息一下?
- 每年的 A 月 B 日提醒我准备朋友的生日礼物?
- ~~cron 的语法好烦啊!~~
- cron 语法辅助工具: <https://crontab.guru>

周期性任务 (3) cron

允许哪些用户使用 cron?

/etc/cron.allow

如果存在该文件，只有这些用户可以用 cron

/etc/cron.deny

如果存在该文件，则除此以外的用户可以用 cron

(root 用户不受任何约束)

如果这两个文件都存在呢?

如何让所有用户使用 cron?

```
$ man 1 crontab
```

周期性任务 (4) anacron

cron 在错过任务后，不会重新执行任务！

- > 在关机期间没必要执行的命令适合直接用 cron
- > 如果要确保命令一定被执行，可以使用 anacron
- > 若安装，anacron 将接管 /etc/cron.daily 等目录下脚本执行
- > anacron 默认在启动时执行，cron 也会每隔一段时间执行它

/etc/anacrontab

	频率	延迟	任务标识符	命令 (command)
示例	@daily	10	backup	/bin/bash /home/user/backup.sh
取值范围	@weekly, @monthly, N (每 N 天)	执行一个任 务前等待分 钟数	用于日志和 时间戳的名 称	(any shell command)

周期性任务 (5)

cron	anacron
独立使用	与 cron 配合使用
时间粒度为分钟	时间粒度为天
关机时不执行任务	下次启动执行关机期间任务

... What about `systemd timers`?

Thanks~