

任务控制

1.

`pgrep sleep` 查名为sleep的进程, `sle`也可以显示sleep出来。

`pkill -af sleep` 终止sleep进程

`-a`: include process ancestor in the match list.

`-f`: match against full argument lists.

2. 如果您希望某个进程结束后再开始另外一个进程, 应该如何实现呢?

`pgrep sleep | wait; ls`

⚠ `#!` contains the process ID of the most recently executed background pipeline.

⚠ `wait` can not wait on its sibling (a child of its parent), can only work only the shell's children.

⚠ each command in a pipeline runs in a separate subshell.

⚠ `kill` 主要用途, 不一定是杀死进程!!!

发送信号到进程 (可以为多个)

列出信号: 返回值 0 成功, 1 失败, 64 部分成功 (当指定了多个进程时)。

`kill -0 $pid`: exit with 1 if process not found, 0 if found.

```
#!/bin/sh-
pidwait () {-
  # kill -0 pid: 不发送任何信号, 但该程序不存在的时候会返回1显示状态不正常。
  while kill -0 $1 # Bash 里返回值 0, while loop才认为是true! -
  do-
    echo "status: $?"-
    sleep 1-
  done-
  # kill -0 $1 返回值为1:-
  ls-
  echo "done! status: $?"-
}-
```

终端多路复用

configuration files: `.tmux.conf`

别名, 执行 `history | awk '{ $1="" ; print substr($0,2) }' | sort | uniq -c | sort -n | tail -n 10` 来获取您最常用的十条命令, 尝试为它们创建别名。

OK

alias dc="cd"

配置文件

<https://pashynskykh.com/posts/chezmoi/>

Using chezmoi to manage dotfiles

用chezmoi同步你的配置文件：

你的电脑

1. **chezmoi init**
2. **chezmoi add <配置文件>** 或 **chezmoi edit <配置文件>; chezmoi apply**
3. **chezmoi cd**
4. **git add .**
5. **git commit -m "msg"**
6. **\$ git remote add origin git@github.com:username/dotfiles.git**
7. **\$ git branch -M main** 改名
8. **\$ git push -u origin main** 推上去

dotfiles

A collection of my personal dotfiles.

Installation

1. Install **chezmoi**

```
snap install chezmoi --classic
```

2. Then restore dotfiles on your machine

```
chezmoi init https://github.com/a2677331/my-dotfiles.git # download all config files inside chezmoi
chezmoi apply                                           # actually place config files under your $USER path
```

新的电脑：

远端设备

1 **ssh-keygen -o -a 100 -t ed25519**

2 用multipass创建虚拟机，然后用**ssh-keygen -o -a 100 -t ed25519**建ssh key，然后登陆。

3 **ssh-copy-id vm**，然后用**ssh vm**登陆

4 使用**python -m http.server 8888**在您的虚拟机中启动一个 Web 服务器并通过本机的**http://localhost:9999** 访问虚拟机上的 Web 服务器 这里我在树莓派上启

动一个服务，并在 MacOS 上通过 ssh 端口转发进行访问。

```
ssh -L 9999:localhost:8888 vm
```

5 使用 `sudo vim /etc/ssh/sshd_config` 编辑 SSH 服务器配置，通过修改 `PasswordAuthentication` 的值来禁用密码验证。通过修改 `PermitRootLogin` 的值来禁用 root 登录。然后使用 `sudo service sshd restart` 重启 ssh 服务器，然后重新尝试。

编辑 SSH 服务器配置: `sudo vim /etc/ssh/sshd_config`

重启 ssh 服务器: `sudo service sshd restart`

#for MacOS

```
sudo launchctl stop com.openssh.sshd
```

```
sudo launchctl start com.openssh.sshd
```

6 (附加题) 在虚拟机中安装 mosh 并启动连接。然后断开服务器/虚拟机的网络适配器。mosh 可以恢复连接吗?

可以，因为 mosh 支出。

7 (附加题) 查看 ssh 的 `-N` 和 `-f` 选项的作用，找出在后台进行端口转发的命令是什么?

```
ssh -fN -L 9999:localhost:8888 vm
```

`-f` 执行命令前先进入后台，后面一定要跟命令! 比如 `-fN`

`-N` Do not execute a remote command. This is useful for just forwarding ports.