Linux系统基础

第四天

陈健

2024年

信号Signal

- sleep命令
  - sleep 10
  - Ctrl-c 中止命令的执行
- 信号signal
  - Ctrl-c → SIGINT信号
  - 查看Linux系统支持的信号类型
    - Kill –l
    - man 7 signal
  - 常见信号
    - SIGINT：中断程序执行
    - SIGQUIT：终止程序执行并生成一个核心转储
    - SIGTERM：通过kill命令发送，终止程序执行
    - SIGHUP：在终端连接被挂断或关闭时发送给进程，终止程序执行
    - SIGSTOP：暂停程序执行
    - SIGCONT：恢复程序执行

捕获SIGINT信号（1）

```
#!/usr/bin/env python
import signal, time

def handler(signum, time):
    print("\nI got a SIGINT, but I am not stopping")

signal.signal(signal.SIGINT, handler)
i = 0
while True:
    time.sleep(.1)
    print("\r{}".format(i), end="")
    i += 1
```

捕获SIGINT信号（2）

```
# ./sigint.py
170^C
I got a SIGINT, but I am not stopping
191^C
I got a SIGINT, but I am not stopping
305^\Quit (core dumped)
```

暂停和后台执行进程（1）

```
# sleep 1000
^Z
[1]+  Stopped                 sleep 1000
```

键入ctrl-z

```
# sleep 1000 &
[2] 329333
```

让命令在后台运行

```
# jobs
[1]+  Stopped                 sleep 1000
[2]-  Running                 sleep 1000 &
```

暂停和后台执行进程（2）

bg %1
[1]+ sleep 1000 &

jobs
[1]- Running sleep 1000 &
[2]+ Running sleep 1000 &

kill -STOP %1

jobs
[1]+ Stopped sleep 1000
[2]- Running sleep 1000 &

暂停和后台执行进程（3）

```
# kill -HUP %1
[1]+  Stopped                 sleep 1000

# jobs
[2]+  Running                 sleep 1000 &

# kill -HUP %2
# jobs
```

暂停和后台执行进程（4）

```
# nohup sleep 1000 &
[1] 331142

# jobs
[1]+  Running                 nohup sleep 1000 &

# kill -HUP %1
# jobs
[1]+  Running                 nohup sleep 1000 &

# kill -KILL %1
# jobs
```

终端多路复用器

- 终端多路复用器允许同时与多个shell会话进行交互，并可以分离当前终端会话以在将来重新连接
- tmux
  - 会话session->窗口window->面板pane
  - 会话
    - tmux new –s NAME 以指定名称开始一个新的会话
    - tmux ls 列出当前所有会话
    - <C-b> d 在tmux中按下该组合键，将当前会话分离
    - tmux a 重新连接最后一个会话，可以通过-t参数指定具体的会话

终端多路复用器

- tmux
  - 窗口
    - <C-b> c 创建一个新的窗口
    - <C-b> p 切换到上一个窗口
    - <C-b> n 切换到下一个窗口
    - <C-b> , 重命名当前窗口
    - <C-b> w 列出当前所有窗口
  - 面板
    - <C-b> " 水平分割
    - <C-b> % 垂直分割
    - <C-b> <方向键> 切换到指定方向的面板
    - <C-b> z 切换当前面板的缩放
    - <C-b> <空格> 在不同的面板布局间切换

别名

```
# 创建常用命令的缩写
alias ll="ls -lh"

# 减少输入字符数
alias gs="git status"
alias gc="git commit"
alias v="vim"

# 打错命令也没关系
alias sl=ls

# 重新定义一些命令的默认行为
alias mv="mv -i"           # -i prompts before overwrite
alias mkdir="mkdir -p"     # -p make parent dirs as needed
alias df="df -h"           # -h prints human readable format

# 别名可以组合使用
alias la="ls -A"
alias lla="la -l"
```

课堂练习

请获取你最常使用的十条命令，并尝试为它们创建别名。

配置文件

- bash配置文件
  - ~/.bashrc
- Git配置文件
  - ~/.gitconfig
- Vim配置文件
  - ~/.vimrc
- SSH配置文件
  - ~/.ssh/config
- tmux配置文件
  - ~/.tmux.conf
- Dotfiles配置资源
  - https://github.com/search?o=desc&q=dotfiles&s=stars&type=Repositories
  - https://github.com/mathiasbynens/dotfiles
  - https://dotfiles.github.io

管理配置文件

将所有配置文件集中放置到一个目录下，并使用版本控制系统进行管理，然后通过脚本将其符号链接到需要的地方

管理配置文件

在本机建立项目仓库，发布到github

~ $ mkdir ~/gits/dotfiles

~ $ git init ~/gits/dotfiles

将本机的配置文件，如 .vimrc/.bashrc/.tmux.conf 等复制进该目录

~ $ ls -a ~/gits/dotfiles

. .. .bashrc .git .tmux.conf .vimrc

其中，". .."分别表示本目录及上级目录，".git"为git仓库的配置文件，其他文件为存放在仓库中的dotfile配置文件

管理配置文件

在另一台机器上将github仓库克隆下来

```
~ $ git clone

~ $ vim autoconfig.sh
~ $ cat autoconfig.sh
#!/bin/bash
files=$(ls -a $1 | grep -E '.[^.]+' |grep -v .git)
# 去掉 ls -a 返回结果中的 ". .. .git"
for file in `echo $files`; do
#创建软链接
        ln -s $1/$file ~/$file
done

# 执行脚本，为dotfiles中的配置文件创建在主目录 ~ 下的软链接
~ $ source autoconfig.sh
```

配置文件可移植性

```
if [[ "$(uname)" == "Linux" ]]; then {do_something}; fi

# 使用和shell相关的配置时先检查当前shell类型
if [[ "$SHELL" == "zsh" ]]; then {do_something}; fi

# 您也可以针对特定的设备进行配置
if [[ "$(hostname)" == "myServer" ]]; then
{do_something}; fi
```

远程服务器

- ssh连接远程服务器
  - ssh test@test.com
- 执行命令
  - ssh test@test.com ls
- SSH密钥
  - ssh-keygen -o -a 100 -t ed25519 -f ~/.ssh/id_ed25519
  - cat .ssh/id_ed25519.pub | ssh username@remote 'cat >> ~/.ssh/authorized_keys'

使用SSH复制文件

- ssh+tee
  - cat localfile | ssh remote_server tee serverfile
- scp
  - scp path/to/local_file remote_host:path/to/remote_file
- rsync
  - rsync –avP . remote_server:test

SSH配置文件

- ~/.ssh/config

```
Host vm
        User foo
        HostName 172.16.1.2
        Port 22
        IdentityFile ~/.ssh/id_ed25519
```

- ssh vm

课后练习

- 完成tmux教程
  - https://hamvocke.com/blog/a-quick-and-easy-guide-to-tmux/
- 学习如何自定义tmux
  - https://www.hamvocke.com/blog/a-guide-to-customizing-your-tmux-conf/

作业4提交方法和截止期限

- 实验报告的文件名命名统一为：学号_lab04.pdf
- 提交截止期限：2024年7月29日零点
- 实验报告通过电子邮件发送给chenj@nju.edu.cn