





第7讲 系统高级管理

主讲：王洪泊





本章学习要点

- 
- 
- (1) **Linux**进程管理
 - (2) 系统启动过程
 - (3) 服务与守护进程
 - (4) 进程调度启动
 - (5) 系统日志管理



以下说法正确的是（ ）。

A

程序本身是一种包含可执行代码的静态文件。

B

多个进程可以并发地调用同一个程序。

C

一个程序可以启动多个进程。

D

每个进程可以有許多子进程。

E

Linux 每一个进程是独立的，有自己的权限及任务。

提交





7.1 Linux进程管理

7.1.1 Linux进程概述

● 进程概念

- 进程由程序产生，是动态的，是一个运行着的、要占用系统运行资源的程序。
- 系统给每一个进程都分配了一个唯一的进程标识符（进程号，简称PID）。

● 进程类型

- 交互进程：在Shell下通过执行程序所产生的进程，可在前台或后台运行。
- 批处理进程：一个进程序列。
- 守护进程：又称监控进程，是指那些在后台运行，并且没有控制终端的进程，通常可以随着操作系统的启动而运行，也可将其称为服务。





7.1 Linux进程管理

7.1.2 查看进程

● 进程参数

- **PID**: 进程号 (Process ID), 用于唯一标识进程。
- **PPID**: 父进程号 (Parent PID), 创建某进程的上一个进程的进程号。
- **USER**: 启动某个进程的用户ID和该用户所属组的ID。
- **STAT**: 进程状态, 一个进程可能处于多种状态, 如运行、等待、停止、睡眠、僵死等。
- **PRIORITY**: 进程的优先级。
- **源占用**: 包括CPU、内存等资源的占用信息。



7.1 Linux进程管理

7.1.2 查看进程

● ps命令

- ps命令是最基本的进程查看命令，可确定有哪些进程正在运行、进程的状态、进程是否结束、进程是否僵死、哪些进程占用了过多的资源等等。
- ps命令最常用的还是监控后台进程的工作情况。
- 最常用的是使用aux选项组合。

```
hong@LAPTOP-DTA58180: ~  
hong@LAPTOP-DTA58180:~$ ps --help  
Usage:  
ps [options]  
  
Try 'ps --help <simple|list|output|threads|misc|all>'  
or 'ps --help <s|l|o|t|m|a>'  
for additional help text.  
  
For more details see ps(1).  
hong@LAPTOP-DTA58180:~$ man ps  
hong@LAPTOP-DTA58180:~$ ps aux  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root         1  0.1  0.0   8896    312 ?        Ssl   16:50    0:00 /init  
root         5  0.0  0.0   8900    216 tty1    Ss    16:50    0:00 /init  
hong        6  0.1  0.0  15076   3560 tty1    S     16:50    0:00 -bash  
hong       51  0.0  0.0  15664   1840 tty1    R     16:53    0:00 ps aux
```

7.1 Linux进程管理

7.1.2 查看进程

●top命令

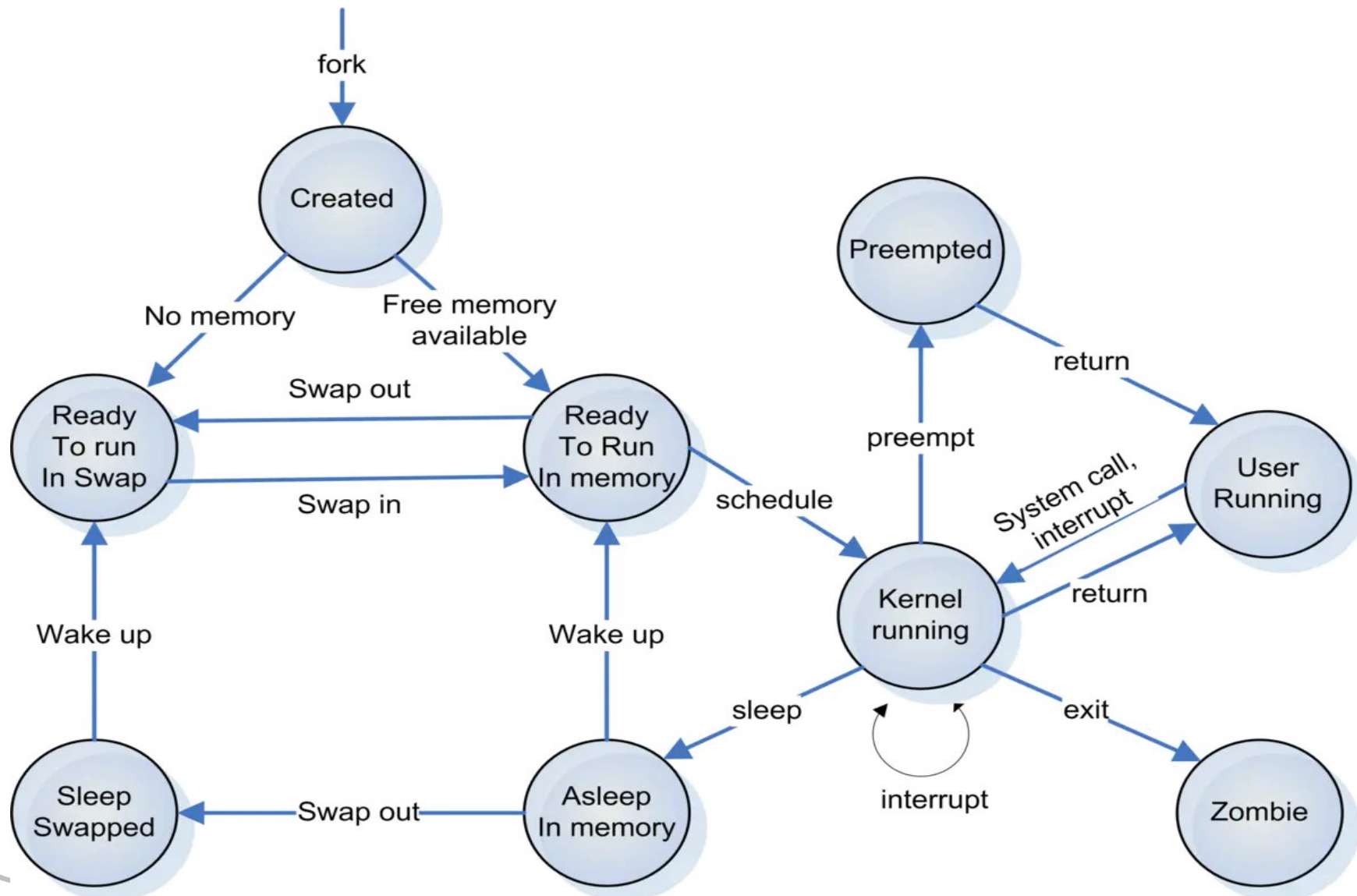
- top命令用于动态显示系统进程信息，可以每隔一短时间刷新当前状态，还提供一组交互式命令用于进程的监控。

```
hong@LAPTOP-DTA5818O: ~  
top - 17:05:02 up 14 min, 0 users, load average: 0.52, 0.58, 0.59  
Tasks: 4 total, 1 running, 3 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 1.7 us, 2.1 sy, 0.0 ni, 95.8 id, 0.0 wa, 0.5 hi, 0.0 si, 0.0 st  
KiB Mem : 8283600 total, 3622008 free, 4425116 used, 236476 buff/cache  
KiB Swap: 25165824 total, 25032288 free, 133536 used. 3717628 avail Mem  


| PID | USER | PR | NI | VIRT  | RES  | SHR  | S | %CPU | %MEM | TIME+   | COMMAND |
|-----|------|----|----|-------|------|------|---|------|------|---------|---------|
| 1   | root | 20 | 0  | 8896  | 296  | 260  | S | 0.0  | 0.0  | 0:00.21 | init    |
| 5   | root | 20 | 0  | 8900  | 204  | 160  | S | 0.0  | 0.0  | 0:00.00 | init    |
| 6   | hong | 20 | 0  | 15076 | 3472 | 3380 | S | 0.0  | 0.0  | 0:00.30 | bash    |
| 52  | hong | 20 | 0  | 15900 | 1932 | 1400 | R | 0.0  | 0.0  | 0:00.83 | top     |


```

Linux进程状态转换图



Process State Diagram

7.1 Linux进程管理

7.1.2 查看进程

● 查看多核CPU命令

- top 命令，然后按数字“1”可监控每个逻辑CPU的状况：

```
hong@LAPTOP-DTA5818O: ~  
top - 17:38:04 up 3 min, 0 users, load average: 0.52, 0.58, 0.59  
Tasks: 4 total, 1 running, 3 sleeping, 0 stopped, 0 zombie  
%Cpu0 :  2.0 us,  5.0 sy,  0.0 ni, 93.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st  
%Cpu1 :  5.6 us,  6.6 sy,  0.0 ni, 86.8 id,  0.0 wa,  1.0 hi,  0.0 si,  0.0 st  
%Cpu2 :  5.3 us,  4.6 sy,  0.0 ni, 90.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st  
%Cpu3 :  3.3 us,  7.0 sy,  0.0 ni, 89.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st  
KiB Mem : 8283600 total, 3128152 free, 4918972 used, 236476 buff/cache  
KiB Swap: 25165824 total, 25028688 free, 137136 used. 3223772 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	8896	288	252	S	0.0	0.0	0:00.23	init
5	root	20	0	8900	200	152	S	0.0	0.0	0:00.00	init
6	hong	20	0	15076	3500	3408	S	0.0	0.0	0:00.29	bash
34	hong	20	0	15900	1900	1368	R	0.0	0.0	0:00.57	top

7.1 Linux进程管理

7.1.2 查看进程

● 查看多核CPU命令

■ mpstat -P ALL

```
hong@LAPTOP-DTA58180: ~  
Creating config file /etc/default/sysstat with new version  
update-alternatives: using /usr/bin/sar.sysstat to provide /usr/bin/sar (sar) in auto mode  
Processing triggers for ureadahead (0.100.0-19.1) ...  
Processing triggers for systemd (229-4ubuntu21.27) ...  
hong@LAPTOP-DTA58180: ~$ mpstat -P ALL  
Linux 4.4.0-18362-Microsoft (LAPTOP-DTA58180) 03/30/2020 _x86_64_ (4 CPU)  
  
05:45:30 PM CPU %usr %nice %sys %iowait %irq %soft %steal %guest %gnice %idle  
05:45:30 PM all 6.41 0.00 7.30 0.00 0.52 0.00 0.00 0.00 0.00 85.77  
05:45:30 PM 0 4.08 0.00 7.04 0.00 0.61 0.00 0.00 0.00 0.00 88.26  
05:45:30 PM 1 9.80 0.00 6.84 0.00 1.29 0.00 0.00 0.00 0.00 82.06  
05:45:30 PM 2 5.96 0.00 8.14 0.00 0.09 0.00 0.00 0.00 0.00 85.82  
05:45:30 PM 3 5.76 0.00 7.19 0.00 0.08 0.00 0.00 0.00 0.00 86.97  
hong@LAPTOP-DTA58180: ~$
```



7.1 Linux进程管理

7.1.2 查看进程

- `cat /proc/cpuinfo | grep name | cut -f2 -d: | uniq -c`
- 能完成啥功能？
- 请小伙伴们弹幕。。。





7.1 Linux进程管理

7.1.2 查看进程

● cat /proc/cpuinfo



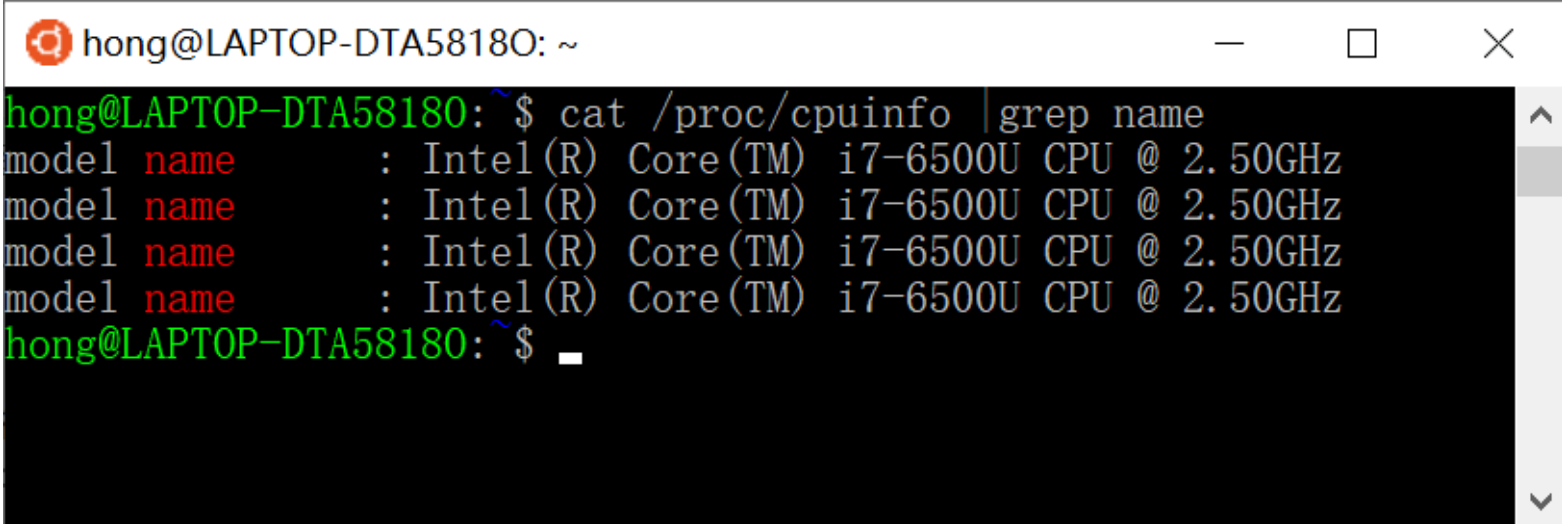
```
hong@LAPTOP-DTA5818O: ~
hong@LAPTOP-DTA5818O: $ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 78
model name     : Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz
stepping       : 3
microcode      : 0xffffffff
cpu MHz        : 2592.000
cache size     : 256 KB
physical id    : 0
siblings       : 4
core id        : 0
cpu cores      : 2
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 6
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi
mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2
ssse3 fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer xsave osxsave avx f16c
rdrand lahf_lm abm 3dnowprefetch fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap c
lflushopt intel_pt ibrs ibpb stibp ssbd
bogomips       : 5184.00
clflush size   : 64
```



7.1 Linux进程管理

7.1.2 查看进程

● `cat /proc/cpuinfo | grep name`



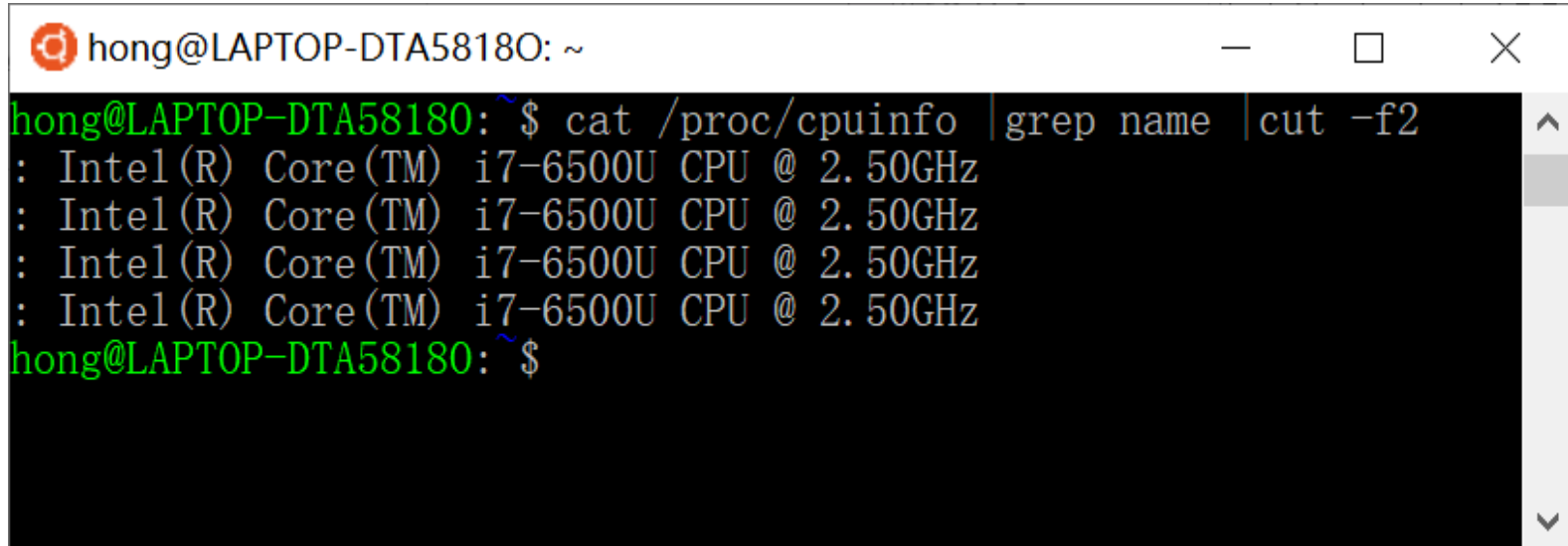
```
hong@LAPTOP-DTA58180: ~  
hong@LAPTOP-DTA58180:~$ cat /proc/cpuinfo | grep name  
model name      : Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
model name      : Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
model name      : Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
model name      : Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
hong@LAPTOP-DTA58180:~$
```



7.1 Linux进程管理

7.1.2 查看进程

● `cat /proc/cpuinfo | grep name | cut -f2`



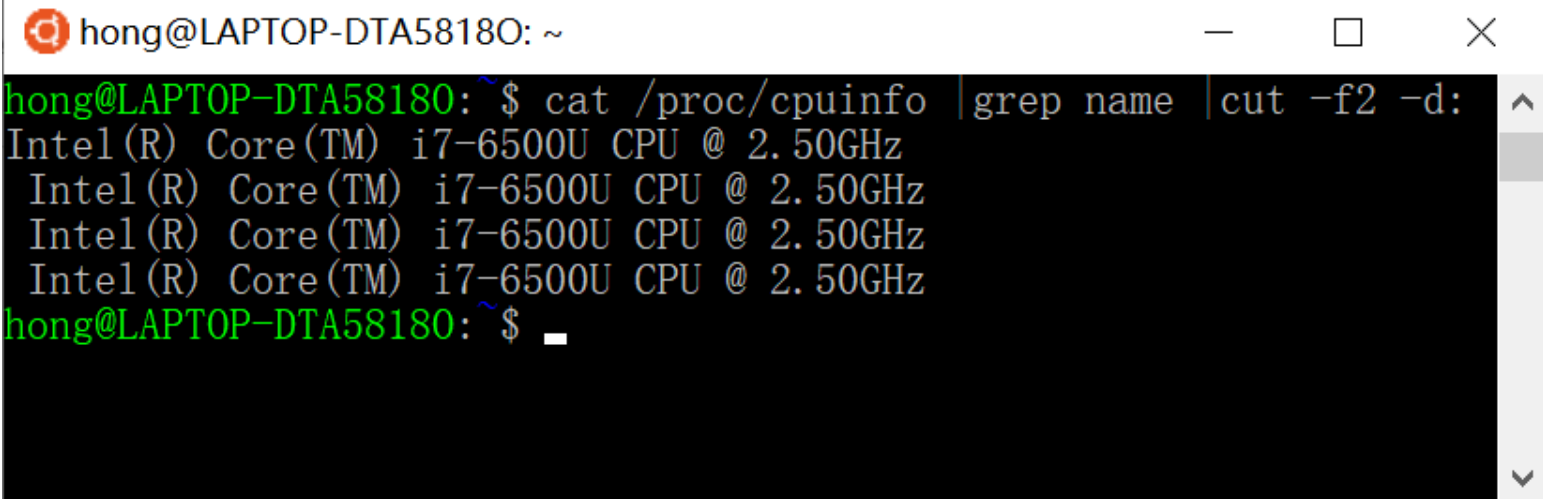
```
hong@LAPTOP-DTA5818O: ~  
hong@LAPTOP-DTA5818O: ~$ cat /proc/cpuinfo | grep name | cut -f2  
: Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
: Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
: Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
: Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
hong@LAPTOP-DTA5818O: ~$
```



7.1 Linux进程管理

7.1.2 查看进程

● `cat /proc/cpuinfo | grep name | cut -f2 -d:`



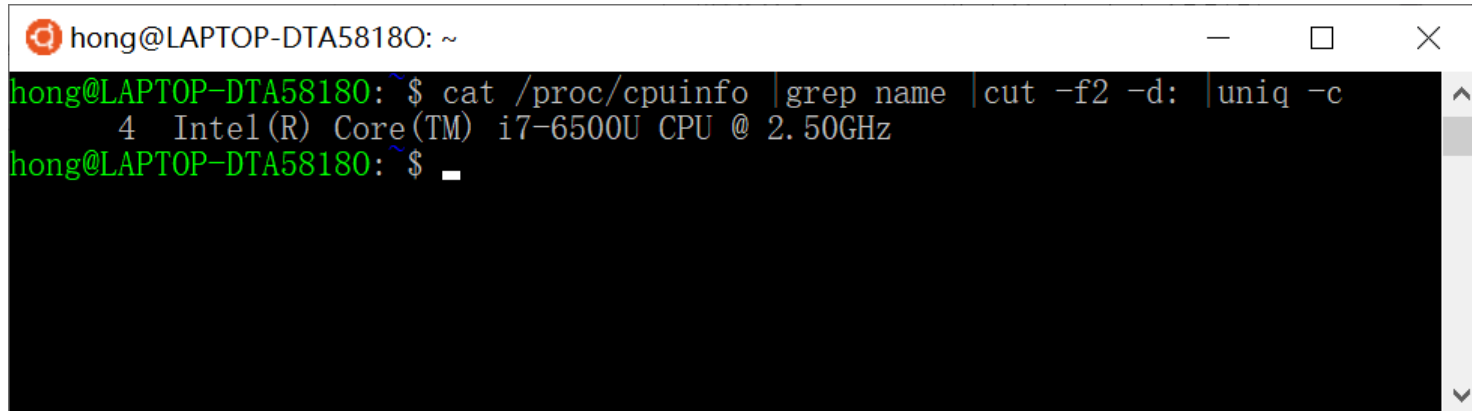
```
hong@LAPTOP-DTA58180: ~  
hong@LAPTOP-DTA58180:~$ cat /proc/cpuinfo | grep name | cut -f2 -d:  
Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
hong@LAPTOP-DTA58180:~$
```



7.1 Linux进程管理

7.1.2 查看进程

● `cat /proc/cpuinfo | grep name | cut -f2 -d: | uniq -c`



```
hong@LAPTOP-DTA58180: ~  
hong@LAPTOP-DTA58180:~$ cat /proc/cpuinfo | grep name | cut -f2 -d: | uniq -c  
      4 Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz  
hong@LAPTOP-DTA58180:~$
```

查看CPU信息（型号）





Linux查看物理CPU个数、核数、逻辑CPU个数

总核数 = 物理CPU个数 X 每颗物理CPU的核数

总逻辑CPU数 = 物理CPU个数 X 每颗物理CPU的核数 X 超线程数

查看物理CPU个数

❖ `cat /proc/cpuinfo| grep "physical id"| sort| uniq| wc -l`

查看每个物理CPU中core的个数(即核数)

❖ `cat /proc/cpuinfo| grep "cpu cores"| uniq`

查看逻辑CPU的个数

❖ `cat /proc/cpuinfo| grep "processor"| wc -l`





7.1 Linux进程管理

7.1.3 Linux进程管理

● 启动进程

- **手动启动：**在Shell命令行下输入要执行的程序来启动一个进程。其启动方式又分为前台启动和后台启动，默认为前台启动。若要在要执行的命令后面跟随一个符号“&”，则为后台启动。
- **调度启动：**事先设置好程序要运行的时间，当到了预设的时间后，系统自动启动程序。

● 进程的挂起及恢复

- 在运行进程的过程中使用<Ctrl>+<Z>组合键可挂起当前的前台作业，将进程转到后台。
- 恢复进程执行有两种选择，
 - 一种是用fg命令将挂起的作业放回到前台执行；
 - 另一种是用bg命令将挂起的作业放到后台执行。





7.1 Linux进程管理

7.1.3 Linux进程管理

● 结束进程的运行

- 使用组合键<Ctrl>+<C>中断前台进程。
- 使用kill命令结束后台进程：

kill [-s,--信号|-p] [-a] 进程号...

● 管理进程的优先级

- 进程优先级可以用nice值表示，范围一般为-20~19，-20为最高优先级，19为最低优先级，系统进程默认的优先级值为0。
- 命令nice用于设置进程的优先级：
nice [-n] [命令 [参数] ...]
- 命令renice用于调整进程的优先级：
renice [优先级] [PID] [进程组] [用户名称或ID]





7.2 系统启动过程

7.2.1 Ubuntu启动过程分析

● 从开机到登录**Ubuntu**系统要经历**4**个主要阶段

- (1) BIOS启动
- (2) 启动引导加载程序
- (3) 装载内核
- (4) 执行init程序实现系统初始化





7.2 系统启动过程

7.2.2 引导加载程序GRUB配置

● GRUB简介

- 在系统启动过程中，从启动引导加载程序开始，到加载内核之前都由GRUB负责。
- GRUB是一种多重操作系统启动管理器。
- 管理员可对GRUB进行配置管理来干预系统的启动。
- 安装Ubuntu系统时会生成GRUB配置文件/boot/grub/grub.conf，通过GRUB引导时会自动读取该配置文件内容，进而显示引导画面。





7.2 系统启动过程

7.2.2 引导加载程序GRUB配置

● 编辑修改/etc/default/grub

设置默认启动菜单项。例如默认从第3个菜单项启动，数字改为2，若改为saved，则默认为上次启动项

```
GRUB_DEFAULT=0
```

#隐藏GRUB菜单，若设置此选项,将隐藏菜单而显示引导画面

```
GRUB_HIDDEN_TIMEOUT=0
```

#设置是否显示倒计时，true表示不显示，屏幕会是空白的

```
GRUB_HIDDEN_TIMEOUT_QUIET=true
```

#设置进入默认启动项的等候时间,若此值为-1将会导致菜单一直显示,直到用户选择

```
GRUB_TIMEOUT="3"
```

```
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
```

#添加内核启动参数

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

#手动添加内核启动参数

```
GRUB_CMDLINE_LINUX="noresume"
```





7.2 系统启动过程

7.2.2 引导加载程序GRUB配置

● 编辑修改/etc/default/grub

```
# Uncomment to disable graphical terminal (grub-pc only)
# 设置是否使用图形界面。去除注释则表示仅使用控制台终端，不使用图形界面
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
# 设置图形界面分辨率
# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true
# Uncomment to disable generation of recovery mode menu entries
# 设置是否创建修复模式菜单项
#GRUB_DISABLE_LINUX_RECOVERY="true"
# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```



7.2 系统启动过程

7.2.3 Ubuntu运行级别

级别	Ubuntu	Redhat
0	关机（Halt）。不要将默认运行级别设置为此级别	关机。不要将默认运行级别设置为此级别
1	单用户（Single）模式。以root身份开启一个虚拟控制台，主要用于管理员维护系统	单用户模式。以root身份开启一个虚拟控制台，主要用于管理员维护系统
2	带显示管理器（GUI）的完整多用户模式	多用户模式，不支持NFS。除不启用网络功能外，与级别3相同
3	带显示管理器（GUI）的完整多用户模式	完整多用户模式。允许所有用户登录，拥有完整的功能，但是以文本模式进入系统保留。用户可自定义环境
4	带显示管理器（GUI）的完整多用户模式	
5	带显示管理器（GUI）的完整多用户模式	X11图形模式。与级别3功能一样，拥有完整功能，以图形界面模式进入系统
6	重启。不要将默认运行级别设置为此级别	重启。不要将默认运行级别设置为此级别



7.2 系统启动过程

7.2.4 init进程与系统运行环境

● System V初始化方式

- System V initialization是以运行级别为核心，依据服务间依赖关系的init方式。
- 使用运行级别和对应的链接文件（位于/etc/rcn.d目录中，n为运行级别，分别链接到/etc/init.d中的init脚本）来启动和关闭系统服务。
- /etc/inittab是相当重要的文件，init进程启动后第一时间找到它，根据它的配置初始化系统，设置系统运行级别及进入各运行级别对应的要执行的命令。





7.2 系统启动过程

7.2.4 init进程与系统运行环境

● Upstart初始化方式

- UpStart是基于事件机制的启动系统，它使用事件来启动和关闭系统服务。
- Upstart不仅能在运行级别改变的时候启动或停止服务，也能在接收到系统发生其他改变的信息的时候启动或停止服务。
- Upstart使用/etc/init目录中的系统服务配置文件来决定系统服务何时启动，何时停止。





7.2 系统启动过程

7.2.4 init进程与系统运行环境

● Ubuntu系统初始化过程

- (1) Upstart执行内部的初始化。
- (2) Upstart触发startup事件，从而触发其余的系统初始化过程。
- (3) init在etc/init目录中搜索指定了“start on startup”的作业，并运行这些作业。
- (4) 由于etc/init目录中的rc-sysinit.conf定义有“start on filesystem”，Upstart将会启动rc-sysinit作业。
- (5) 调用telinit进入了设定的运行级别，runlevel事件发生。





7.3 服务与守护进程管理

7.3.1 服务与守护进程的概念

- 在Linux系统中有些程序在启动之后持续在后台运行，等待用户或其他应用程序调用，此类程序就是服务（**service**）。
- 大多数服务都是通过守护进程（**daemon**）实现的。
- 守护进程是服务的具体实现。
- Ubuntu系统启动时会自动启动很多守护进程（系统服务）。
- 守护进程按照功能可以区分为系统守护进程与网络守护进程。





7.3 服务与守护进程管理

7.3.2 Linux网络服务定义文件/etc/services

- Linux使用Internet网络服务文件/etc/services来定义网络服务名和它们对应使用的端口号及协议。
- /etc/services文件中的每一行对应一种服务，由4个字段组成，分别表示服务名、所用端口号、协议名和别名。
- Linux系统的端口号的范围为0~65535，不同范围有不同的意义，具体分为以下3个范围。
 - 公认端口（端口0到1023）：用于服务和应用程序。
 - 已注册端口（端口1024到49151）：分配给用户进程或应用程序。
 - 动态或私有端口（端口49152到65535）：又称为临时端口，往往在开始连接时被动态分配给客户端应用程序。



7.3 服务与守护进程管理

7.3.3 Linux服务启动脚本

● 了解Ubuntu的服务启动脚本

- 每个服务都有相应的服务启动脚本，Ubuntu中将其保存在 `/etc/init.d` 目录下。
- 脚本文件的名称使用守护进程名或服务名称，常用服务的服务名（包括服务别名）通常在 `/etc/services` 中列出。
- 每一个服务启动脚本可用于实现启动服务、重启服务、停止服务和查询服务等功能。





7.3 服务与守护进程管理

7.3.3 Linux服务启动脚本

● 服务启动脚本与运行级别

- Linux使用rc脚本统一管理每个服务的脚本程序。
- Ubuntu将各运行级别对应的脚本文件存放在/etc/rcn.d目录中。
- /etc/rcn.d目录中存放的是指向/etc/init.d目录中脚本程序的符号链接，而实际的脚本程序保存在/etc/init.d目录中。
- 符号链接的命名规则比较特别：
 - 如果脚本是用来启动一个服务的，其符号链接的名称就以字母S打头。
 - 如果脚本是用来停止一个服务的，其符号链接的名称就以字母K打头。
 - 字母S或K之后是一个数字，表示脚本的执行顺序。
 - 执行顺序数字后通常是符号链接所指向的脚本文件的名称。





7.3 服务与守护进程管理

7.3.3 Linux服务启动脚本

● 使用**update-rc.d**管理启动脚本

- Ubuntu可以使用**update-rc.d**命令管理启动脚本，修改服务的启动选项。
- 使用**update-rc.d**的优势很明显，它会自己移除、增加对**/etc/init.d/**内的链接。
- 使用以下命令从所有的运行级别中删除指定服务**cups**的启动项。
`sudo update-rc.d -f cups remove`
- 使用以下命令往所有的运行级别中添加指定服务**cups**的启动项。
`sudo update-rc.d -f cups defaults`





7.3 服务与守护进程管理

7.3.4 执行服务启动脚本

- 使用**Linux**服务启动脚本管理服务

- 使用服务启动脚本即可实现启动服务、重启服务、停止服务和查询服务等功能：

/etc/init.d/服务启动脚本名 {start|stop|status|restart|reload|force-reload}

- 使用**service**命令简化服务管理

- **service**命令功能和参数与使用服务启动脚本相同：

service 服务启动脚本名 {start|stop|status|restart|reload|force-reload}





7.3 服务与守护进程管理

7.3.5 配置服务启动状态

- 使用**sysv-rc-conf**配置服务启动状态

- 查看当前各服务的启动状态：

`sysv-rc-conf --list`

- 使用该工具启动或关闭某项指定服务：

`sysv-rc-conf 服务名 <on|off>`

- 设置指定运行级别中服务的启动状态：

`sysv-rc-conf --level <运行级别列表> 服务名 <on|off>`

- 使用**update-rc.d**配置服务启动状态

- 使用update-rc.d在指定运行级别中启动或关闭某项服务：

`update-rc.d [-f] <服务名> disable|enable [S|2|3|4|5]`





7.4 进程的调度启动——自动化任务配置

7.4.1 使用cron工具安排周期性任务

- 使用配置文件/etc/crontab定义系统级周期性任务

- cron主要使用配置文件/etc/crontab来管理系统级任务调度。
- 该配置文件共有4行任务定义，每行格式为：
分钟（m） 小时（h） 日期（dom） 月份（mon） 星期（dow） 用户身份（user） 要执行的命令（command）
- /etc/crontab文件中并没有定义要执行的具体作业，而是在最后一个字段中设置了可执行文件目录，/etc/cron.hourly、/etc/cron.daily、/etc/cron.weekly和/etc/cron.monthly分别表示每小时、每日、每周和每月执行要执行任务的目录。

- 在etc/cron.d目录中定义个别的周期性任务

- 要为计划任务指定其他时间点，则可以考虑在/etc/cron.d/目录中添加自己的配置文件，格式同/etc/crontab，文件名可以自定义。



7.3 服务与守护进程管理

7.3.5 配置服务启动状态

- 使用**crontab**命令为普通用户定制任务调度

- 普通用户只能使用**crontab**命令创建和维护自己的cron配置文件:

crontab [-u 用户名] [-e | -l | -r]

```
zxp@LinuxPC1: ~  
GNU nano 2.2.6 文件: /tmp/crontab.H8GW2l/crontab  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow  command  
^G 求助      ^O 写入      ^R 读档      ^Y 上页      ^K 剪切文字  ^C 光标位置  
^X 离开      ^J 对齐      ^W 搜索      ^V 下页      ^U 还原剪切  ^T 拼写检查
```



7.4 进程的调度启动——自动化任务配置

7.4.2 使用at和batch工具安排一次性任务

● 概述

- 使用**at**在指定时间内调度一次性任务。
- **batch**用于在系统平均载量降到0.8以下时执行一次性的任务
- 两个工具都由**at**软件包提供，由**at**服务支持。

● 配置**at**作业的步骤

- (1) 在命令行中执行**at**命令进入作业设置状态。
- (2) 出现**at>**提示符，进入命令编辑状态，设置要执行的命令或脚本。可指定多条命令，每输入一条命令，按<Enter>键。
- (3) 需要结束时按<Ctrl>+<D>组合键退出。
- (4) 可根据需要执行命令**atq**查看等待运行（未执行）的作业。
- (5) 如果**at**作业需要取消，可以在**atrm**命令后跟**atq**命令输出的作业号来删除该**at**作业。





7.5 系统日志管理

7.5.1 配置系统日志

● 系统日志相关文件

- Ubuntu的日志配置文件为/etc/rsyslog.conf。
- 采用rsyslog的Ubuntu的rsyslog.conf配置文件将所有的配置文件放置在/etc/rsyslog.d目录中，默认的etc/rsyslog.d/50-default.conf，可以定制该文件来实现系统日志的配置，如记录日志的信息来源、信息类型以及保存位置。
- 配置文件中每一行都代表一条设置值，语法如下。

信息来源.优先级	处理方式
----------	------





7.5 系统日志管理

7.5.1 配置系统日志

● 日志文件定义规则——信息来源

- 信息来源定义日志记录来自哪个子系统。

信息来源	说明	信息来源	说明
authpriv	安全/授权	mail	电子邮件系统
cron	at或cron定时执行任务	news	网络新闻系统
daemon	守护进程	syslog	syslogd内部
ftp	ftp守护进程	user	一般用户级别
kern	内核	uucp	UUCP系统
lpr	打印系统	localN	保留



7.5 系统日志管理

7.5.1 配置系统日志

- 日志文件定义规则——优先级
 - 优先级代表信息的紧急程度。

信息来源	说明	信息来源	说明
debug	调试排错信息，仅对程序开发人员有用	err	一般的错误信息
info	一般信息，可以忽略	crit	关键状态信息
notice	正常提示信息	alert	需特别注意的警报信息，一般要迅速更正
warn	可能是有问题的警告信息	emerg	最严重，紧急状况，一般是系统不可用





7.5 系统日志管理

7.5.1 配置系统日志

● 日志文件定义规则——处理方式

- **syslog**的处理动作用来定义如何处理接收到的信息，通常是将信息发往何处。主要有以下几种处理方式。
 - 将信息存储到指定文件。
 - 将信息发送到指定设备。
 - 将信息发给某个用户。
 - 将信息发送到命名管道。
 - 将信息发送到远程主机。





7.5 系统日志管理

7.5.2 查看和管理系统日志内容

● 日志文件定义规则——处理方式

- 大部分的日志文件集中存储在/var/log目录。
- 系统日志服务产生的记录文件的每一行就是一条信息，每一行包含的字段主要有：

信息发生的日期、时间、主机、产生信息的软件、软件或者软件组件的名称（可以省略）、PID（进程标识符，可以省略）、信息内容。

- 使用echo命令清空日志文件的内容，命令格式如下所示。
echo > 日志文件





习题

1. **Linux 进程有哪几种类型？**
2. **简述进程的手动启动和调度启动。**
3. **Ubuntu系统启动经过哪四个阶段？**
4. **Ubuntu的运行级别有哪几种？与Redhat等Linux版本有何区别？**
5. **简述什么是守护进程？**
6. **简述Ubuntu系统服务启动脚本与运行级别的关系。**
7. **简述Ubuntu系统日志配置文件的基本格式。**
8. **如何执行服务启动脚本？如何配置服务启动状态？**

