

实验题目：Linux 进程调度与系统监视

姓名：王波

学号：19122557

实验日期：2021 年 9 月 28 日

Linux 进程调度与系统监视

【实验要求】

1. 熟练掌握手工启动前后台作业的方法。
2. 熟练掌握进程与作业管理的相关 Shell 命令。
3. 掌握 at 调度和 cron 调度的设置方法。
4. 了解进行系统性能监视的基本方法。

【实验环境】

虚拟机中 Linux 环境（CentOS7 镜像）

【实验步骤、结果截图】

（一）作业和进程的基本管理

【操作要求 1】先在前台启动 vi 编辑器并打开 f4 文件，然后挂起，最后在后台启动一个查找 inittab 文件的 find 作业，find 的查找结果保存到 f5。

① 以超级用户（root）身份登录到 RHEL Server 5 字符界面。

```
master@master:/home/master
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[ master@master ~]$ su
密码：
[ root@master master]#
```

② 输入命令“vi f4”，在前台启动 vi 文本编辑器并打开 f4 文件。

```
[ root@master master]# vi f4
```

③ 按下 Ctrl+Z 组合键，暂时挂起“vi f4”作业，屏幕显示该作业的作业号。

```
[ root@master master]# vi f4
[1]+ 已停止          vi f4
[ root@master master]#
```

④ 输入命令“find / -name inittab > f5 &”，启动一个后台作业，如下所示。在显示作业号的同时还显示进程号。

```
[ root@master master]# find / -name inittab >f5 &
[2] 7862
```

【操作要求 2】查看当前作业、进程和用户信息，并对作业进行前后台切换。

① 输入命令“jobs”，查看当前系统中的所有作业

```
jobs
[1]+ 已停止                  vi f4
[2]- 退出 1                  find / -name inittab > f5
```

由此可知“vi f4”作业的作业号为 1，已经停止。“find / -name inittab > f5 &”作业的作业号为 2，已退出（因为我的 centos 版本和指导书上的不一样，该作业无法在后台运行，因为找不到文件只能退出）。

```
[root@master master]# find: '/proc/7863': 没有那个文件或目录
find: '/proc/7865': 没有那个文件或目录
```

无奈之下，更换了镜像，在 Ubuntu 虚拟机中测试了一下，但还是会报错，网上查阅了一下，发现是 FUSE 文件系统和权限不配的问题，全局搜索时碰到它就会退出报错。但其实这个目录是空的，查不查都没关系，故选择删除，最终该命令可以顺利执行。

```
root@ubuntu:/home/master-1/Desktop# find: '/run/user/1000/gvfs': Permission denied
jobs
```

```
root@ubuntu:/home/master-1/Desktop# umount /run/user/1000/gvfs
root@ubuntu:/home/master-1/Desktop# rm -rf /run/user/1000/gvfs
```

```
root@ubuntu:/home/master-1/Desktop# jobs
[1]+  Stopped                  vi f4
[2]-  Done                     find / -name inittab > f5
```

②输入命令“fg 2”，将“find / -name inittab > f5 &”作业切换到前台。屏幕显示出“find / -name inittab > f5”命令，并执行此命令。稍等片刻，作业完成后屏幕再次出现命令提示符。

③输入命令“cat f5”，查看“find / -name inittab > f5”命令的执行结果

```
[root@master master]# cat f5
/etc/inittab
```

④再次输入命令“jobs”，可发现当前系统中的只有一个已停止的作业“vi f4”。

```
root@ubuntu:/home/master-1/Desktop# jobs
[1]+  Stopped                  vi f4
root@ubuntu:/home/master-1/Desktop#
```

⑤输入命令“kill -9 %1”，终止“vi f4”作业。

```
root@ubuntu:/home/master-1/Desktop# kill -9 %1

[1]+  Stopped                  vi f4
```

使用 kill 命令时“-9”选项可强制性中止进程或作业。

⑥稍等片刻，输入命令“jobs”，查看当前没有任何作业。

```
root@ubuntu:/home/master-1/Desktop# jobs
root@ubuntu:/home/master-1/Desktop# jobs
root@ubuntu:/home/master-1/Desktop#
```

⑦输入命令“ps -l”，查看进程的相关信息，显示出的信息类似如下信息

```
root@ubuntu:/home/master-1/Desktop# ps -l
F S  UID      PID     PPID  C  PRI  NI ADDR  SZ  WCHAN  TTY          TIME CMD
4 S   0       2106    2100  0   80   0  -   5014 do_wai pts/0    00:00:00 su
4 S   0       2107    2106  0   80   0  -   4684 do_wai pts/0    00:00:00 bash
4 R   0       2201    2107  0   80   0  -   5013 -      pts/0    00:00:00 ps
root@ubuntu:/home/master-1/Desktop#
```

主要输出项的含义为：

S（State）：进程状态，其中 R 表示运行状态；S 表示休眠状态；T 表示暂停或终止状态；Z 表示僵死状态。

UID（User ID）：进程启动者的用户 ID。

PID（Process ID）：进程号。

PPID (Parents Process ID) : 父进程的进程号。

NI (Nice) :进程的优先级值。

SZ (Size) :进程占用内存空间的大小, 为 KB 为单位。

TTY (Terminal) :进程所在终端的终端号, 其中桌面环境的终端窗口表示为 pts/0, 字符界面的终端号为 tty1~tty6。

TIME :进程已运行的时间。

CMD (Command) :启动该进程的 Shell 命令。

⑧ 输入命令“who -H”, 查看用户信息。

```
root@ubuntu:/home/master-1/Desktop# who -H
NAME      LINE      TIME          COMMENT
master-1  :0        2021-10-13 22:26 (:0)
root@ubuntu:/home/master-1/Desktop#
```

(二) at 进程调度

【操作要求 1】设置一个调度, 要求在 2008 年 1 月 1 日 0 时, 向所有用户发送新年快乐的问候

①超级用户输入命令“at 00:00 01012008”, 设置 2008 年 1 月 1 日 0 时执行的 at 调度的内容。

```
[root@master master]# at 00:00 01012008
at: refusing to create job destined in the past
[root@master master]#
```

因无法创建日期已过的调度, 故修改为 2022 年

②屏幕出现 at 调度的命令提示符“at>”, 输入“wall Happy New Year!”, 向所有用户发送消息。

③光标移动到“at>”提示符的第三行, 按下 Ctrl+D 组合键结束输入。根据调度设置的时间, 最后显示出作业号和将要运行的时间。

```
[root@master master]# at 00:00 01012022
at> wall Happy New Year!
at> <EOT>
job 3 at Sat Jan 1 00:00:00 2022
[root@master master]#
```

【操作要求 2】设置一个调度, 要求 5 分钟后向所有用户发送系统即将重启的消息, 并在 2 分钟后重新启动计算机。

① 超级用户输入命令“at now +5 minutes”, 设置 5 分钟后执行的 at 调度的内容。

② 屏幕出现 at 调度的命令提示符“at>”, 输入“wall please logout; the computer will restart.”, 向所有用户发送消息。

③在“at>”提示符的第二行输入“shutdown -r +2”, 系统 2 分钟后将重新启动。“shutdown -r +2”命令与“reboot +2”命令效果相同, 都是在 2 分钟后重新启动。

④光标移动到“at>”提示符的第三行, 按下 Ctrl+D 组合键结束输入。最后显示作业号和运行时间。

```
[root@master master]# at now+5minutes
at> wall please logout; the computer will restart
at> shutdown -r +2
at> <EOT>
job 4 at Thu Oct 14 14:24:00 2021
[root@master master]#
```

【操作要求 3】查看所有的 at 调度, 并删除 08 年 1 月 1 日执行的调度任务。

①输入“atq”命令, 查看所有的 at 调度, 显示出作业号、将在何时运行以及 at 调度

的设定者

```
[root@master master]# atq
3      Sat Jan  1 00:00:00 2022 a root
4      Thu Oct 14 14:24:00 2021 a root
[root@master master]#
```

②输入“atrm 1”命令删除作业号为 3 的 at 调度，并再次输入“atq”命令查看剩余的所有 at 调度内容。

```
[root@master master]# atrm 3
[root@master master]# atq
4      Thu Oct 14 14:24:00 2021 a root
[root@master master]#
```

③ 5 分钟后系统将自动运行作业号为 2 的 at 调度内容。先向所有用户发送消息，然后再等 2 分钟重新启动。

（三）cron 进程调度

注：为立即查看调度结果，可以切换到特权用户修改系统的时间

【操作要求 1】helen 用户设置 crontab 调度，要求每天上午 8 点 30 份查看系统的进程状态，并将查看结果保存于 ps.log 文件

① 以普通用户 helen 登录，并输入命令“crontab -e”，新建一个 crontab 配置文件。

```
[master@master ~]$ crontab -e
```

② 屏幕出现 vi 编辑器，按下“i”，进入输入模式，输入“30 8 * * * ps >ps.log”。

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
30 8 * * * ps>ps.log
~
```

③ 按下 Esc 键退出 vi 的文本输入模式，并按下“:”键切换到最后行模式，输入“wq”，保存并退出编辑器，显示“crontab: installing new crontab”信息。

```
[master@master ~]$ crontab -e
no crontab for master - using an empty one
crontab: installing new crontab
[master@master ~]$
```

④ 输入命令“crontab -l”，查看 helen 用户的 cron 调度内容。

```
[master@master ~]$ crontab -l
30 8 * * * ps>ps.log
[master@master ~]$
```

⑤为立即查看到 crontab 调度的结果，切换为超级用户，并适当修改系统时间，如修改为 8 点 29 分。最后退回到 helen 用户。

```
[master@master ~]$ su
密码:
[root@master master]# date 11200829
2021年 11月 20日 星期六 08:29:00 CST
[root@master master]# exit
exit
[master@master ~]$
```

⑥等待 1 分钟后，查看 ps.log 文件的内容，如果显示出正确的内容，那么说明 crontab 调度设置成功。

```
ps.log
PID TTY      TIME CMD
1744 ?        00:00:00 gnome-keyring-d
1758 ?        00:00:00 gnome-session-b
1768 ?        00:00:00 dbus-launch
1769 ?        00:00:00 dbus-daemon
1796 ?        00:00:00 imsettings-daem
1800 ?        00:00:00 gvfsd
1805 ?        00:00:00 gvfsd-fuse
1887 ?        00:00:00 ssh-agent
1905 ?        00:00:00 at-spi-bus-laun
1910 ?        00:00:00 dbus-daemon
1915 ?        00:00:00 at-spi2-registr
1942 ?        00:00:08 gnome-shell
1952 ?        00:00:00 nlscaudio
```

【操作要求 2】helen 用户添加设置 crontab 调度, 要求每三个月的 1 号零时查看正在使用的用户列表。

- ① 再次输入命令 “crontab -e”, 出现 vi 编辑器, 按下 “i”, 屏幕进入文本输入模式。
- ② 在原有内容之后, 另起一行, 输入 “0 0 * */3 * who >who.log”。

```
30 8 * * * ps>ps.log
0 0 * */3 * who>who.log
```

- ③ 最后保存并退出 vi 编辑器。
- ④ 为立即查看到 crontab 调度的结果, 切换为超级用户, 并适当修改系统时间, 如修改为 3 月 31 日 23 点 59 分。最后退回到 helen 用户。

```
[master@master ~]$ su
密码:
[root@master master]# date 03312359
2021年 03月 31日 星期三 23:59:00 CST
[root@master master]# exit
exit
[master@master ~]$
```

- ⑤ 等待 1 分钟后, 查看 who.log 文件的内容, 如果显示出正确的内容, 那么说明新增的 crontab 调度设置成功。

```
who.log
master :0      2021-10-14 11:00 (:0)
master pts/0   2021-10-14 11:01 (:0)
```

【操作要求 3】查看 cron 调度内容, 最后删除此调度。

- ① 输入命令 “crontab -l”, 查看 cron 调度内容。

```
[master@master ~]$ crontab -l
30 8 * * * ps>ps.log
0 0 * */3 * who>who.log
[master@master ~]$
```

- ② 输入命令 “crontab -r”, 删除 cron 调度内容。
- ③ 再次输入命令 “crontab -l”, 此时无 cron 调度内容。

```
[master@master ~]$ crontab -r
[master@master ~]$ crontab -l
no crontab for master
```

(四) 系统性能监视

【操作要求 1】利用 Shell 命令监视系统性能

① 输入命令“top”，屏幕动态显示 CPU 利用率、内存利用率和进程状态等相关信息

```
top - 00:05:25 up 3:54, 2 users, load average: 0.00, 0.03, 0.05
Tasks: 213 total, 2 running, 211 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.1 us, 0.3 sy, 0.0 ni, 95.6 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
Mem: 3107632 total, 202128 free, 286016 used, 0 buffers, 0 cache
ali-Linux-2021.3-vmware-amd64
虚拟机镜像(kali-linux-2021.3-vmware-amd64.vmx)Kali-Linux-2021.3-vmware-amd64.vmx
```

| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|-------|--------|----|-----|---------|--------|-------|---|------|------|---------|-----------------|
| 1942 | master | 20 | 0 | 3107632 | 202128 | 48832 | S | 3.3 | 20.3 | 1:47.00 | gnome-shell |
| 1389 | root | 20 | 0 | 389764 | 91856 | 35588 | S | 1.0 | 9.2 | 0:20.83 | X |
| 5041 | master | 20 | 0 | 776612 | 27308 | 11976 | S | 0.7 | 2.7 | 0:04.77 | gnome-terminal- |
| 2268 | master | 20 | 0 | 612124 | 8452 | 2860 | S | 0.3 | 0.8 | 0:12.94 | vmtoolsd |
| 12861 | root | 20 | 0 | 0 | 0 | 0 | R | 0.3 | 0.0 | 0:00.54 | kworker/0:0 |
| 1 | root | 20 | 0 | 202156 | 4980 | 2620 | S | 0.0 | 0.5 | 0:03.61 | systemd |
| 2 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.01 | kthreadd |
| 4 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kworker/0:0H |
| 6 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.37 | ksoftirqd/0 |

② 按下 M 键，所有进程按照内存使用率排列

```
top - 00:05:58 up 3:55, 2 users, load average: 0.07, 0.05, 0.05
Tasks: 213 total, 1 running, 212 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.7 us, 0.3 sy, 0.0 ni, 97.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 87.7/995672  [|||||]
KiB Swap: 10.8/2097148 [|||||]
```

| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|-------|--------|----|-----|---------|--------|-------|---|------|------|---------|-----------------|
| 1942 | master | 20 | 0 | 3107632 | 202140 | 48832 | S | 2.0 | 20.3 | 1:47.42 | gnome-shell |
| 1389 | root | 20 | 0 | 389764 | 91856 | 35588 | S | 0.7 | 9.2 | 0:20.94 | X |
| 5041 | master | 20 | 0 | 776612 | 27308 | 11976 | S | 0.3 | 2.7 | 0:04.81 | gnome-terminal- |
| 14507 | master | 20 | 0 | 162112 | 2372 | 1580 | R | 0.3 | 0.2 | 0:00.08 | top |
| 1 | root | 20 | 0 | 202156 | 4980 | 2620 | S | 0.0 | 0.5 | 0:03.62 | systemd |
| 2 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.01 | kthreadd |
| 4 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kworker/0:0H |
| 6 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.37 | ksoftirqd/0 |
| 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 | 0:00.97 | rcu_sched |
| 10 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | lru-add-drain |
| 11 | root | rt | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.08 | watchdog/0 |
| 13 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kdevtmpfs |
| 14 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | netns |

③ 按下 T 键，所有进程按照执行时间排列

```
top - 00:06:10 up 3:55, 2 users, load average: 0.03, 0.04, 0.03
Tasks: 212 total, 1 running, 211 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.7/0.7 4 [||||]
KiB Mem : 87.7/995672  [|||||]
CentOS 7 64 位
/Users/23800/Documents/Virtual Machines/CentOS 7 64 位/CentOS 7 64 位.vmx
```

| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|------|--------|----|-----|---------|--------|-------|---|------|------|---------|-----------------|
| 1942 | master | 20 | 0 | 3107632 | 202160 | 48832 | S | 3.0 | 20.3 | 1:47.72 | gnome-shell |
| 1389 | root | 20 | 0 | 389764 | 91856 | 35588 | S | 1.0 | 9.2 | 0:21.02 | X |
| 5041 | master | 20 | 0 | 776612 | 27308 | 11976 | S | 0.3 | 2.7 | 0:04.85 | gnome-terminal- |
| 1 | root | 20 | 0 | 202156 | 4980 | 2620 | S | 0.0 | 0.5 | 0:03.62 | systemd |
| 2 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.01 | kthreadd |
| 4 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kworker/0:0H |
| 6 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.37 | ksoftirqd/0 |
| 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 | 0:00.97 | rcu_sched |
| 10 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | lru-add-drain |

④最后按下 C 键，恢复按照 CPU 使用率排列所有进程。

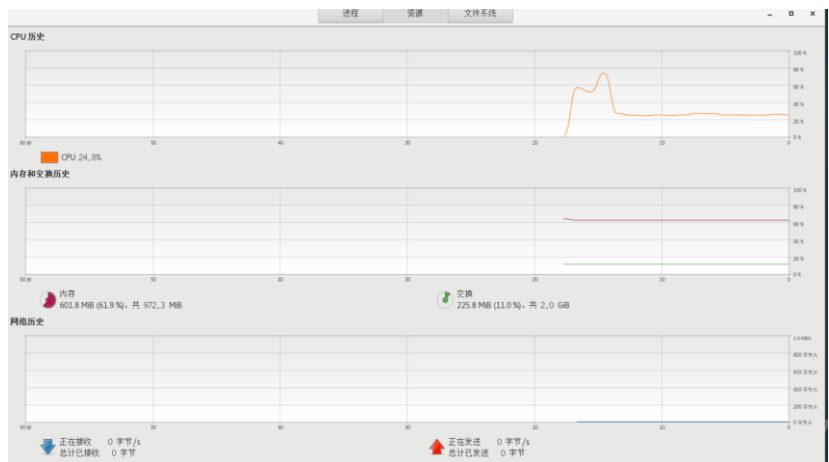
```
%Cpu(s): 87.5/0.0 7 [||||]
KiB Mem : 87.5/995672  [|||||]
KiB Swap: 10.8/2097148 [|||||]
```

| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|-------|--------|----|-----|---------|--------|-------|---|------|------|---------|--|
| 1942 | master | 20 | 0 | 3107632 | 202188 | 48832 | S | 2.7 | 20.3 | 1:48.27 | /usr/bin/gnome-shell |
| 1389 | root | 20 | 0 | 389764 | 91856 | 35588 | S | 1.0 | 9.2 | 0:21.18 | /usr/bin/X :0 - background none - noreset - audit 4 - verbose - auth /run/gdm/auth-1 |
| 290 | root | 20 | 0 | 0 | 0 | 0 | S | 0.3 | 0.0 | 0:04.66 | [xfssaild/sda3] |
| 5041 | master | 20 | 0 | 776612 | 27308 | 11976 | S | 0.3 | 2.7 | 0:04.92 | /usr/libexec/gnome-terminal-server |
| 12861 | root | 20 | 0 | 0 | 0 | 0 | S | 0.3 | 0.0 | 0:00.59 | [kworker/0:0] |
| 14507 | master | 20 | 0 | 162140 | 2372 | 1580 | R | 0.3 | 0.2 | 0:00.16 | top |
| 1 | root | 20 | 0 | 202156 | 4980 | 2620 | S | 0.0 | 0.5 | 0:03.62 | /usr/lib/systemd/systemd -- switched-root --system --deserialize 22 |
| 2 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.01 | [kthreadd] |
| 4 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | [kworker/0:0H] |
| 6 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.37 | [ksoftirqd/0] |
| 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 | 0:00.97 | [rcu_sched] |
| 10 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | [lru-add-drain] |
| 11 | root | rt | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.08 | [watchdog/0] |
| 13 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | [kdevtmpfs] |
| 14 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | [netns] |
| 15 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.01 | [khungtaskd] |
| 16 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | [writeback] |
| 17 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | [kintegrityd] |
| 18 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | [bioset] |

⑤ 按下 CTRL+C 组合键结束 top 命令

【操作要求 2】利用「系统监视器」工具监视 CPU 使用情况。

- ① 启动 GNOME 桌面环境，依次单击「系统」菜单=>「管理」=>「系统监视器」，打开「系统监视器」窗口。
- ②自动显示「资源」选项卡，查看当前 CPU、内存和交换分区、网络历史的使用情况



- 【操作要求 3】利用「系统监视器」查看当前所有的进程，要求显示出启动进程的用户。
- ① 在「系统监视器」窗口单击「进程列表」选项卡，默认显示当前用户启动的所有进程。单击「查看」菜单，选中「所有的进程」单选按钮，并选中「依赖关系」复选框，则显示系统中所有的进程

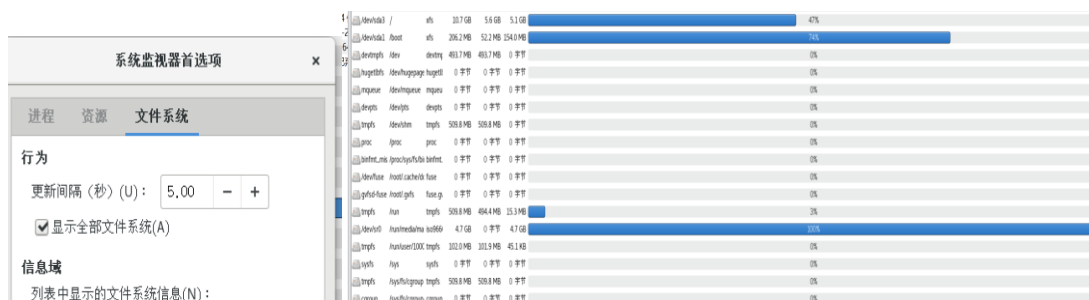
| 进程名 | 用户 | % CPU | ID | 内存 | 磁盘读取总计 | 磁盘写入总计 | 磁盘读取 | 磁盘写入 | 优先级 |
|-----------------|------|-------|------|-----------|--------|--------|------|------|-----|
| kthreadd | root | 0 | 2 | 不适用 | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| systemd | root | 0 | 1 | 2.3 MiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| abrt | root | 0 | 607 | 4.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| abrt-watch-log | root | 0 | 613 | 40.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| accounts-daemon | root | 0 | 578 | 556.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| alsactl | root | 0 | 627 | 4.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 非常低 |
| atd | root | 0 | 1002 | 44.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |

- ② 单击「编辑」菜单中的「首选项」，弹出「系统监视器首选项」对话框。在「进程」选项卡，选中「进程域」栏的「用户」复选框，要求显示出启动进程的用户，如

| 进程名 | 用户 | % CPU | ID | 内存 | 磁盘读取总计 | 磁盘写入总计 | 磁盘读取 | 磁盘写入 | 优先级 |
|-----------------|------|-------|------|-----------|--------|--------|------|------|-----|
| kthreadd | root | 0 | 2 | 不适用 | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| systemd | root | 0 | 1 | 2.3 MiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| abrt | root | 0 | 607 | 4.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| abrt-watch-log | root | 0 | 613 | 40.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| accounts-daemon | root | 0 | 578 | 556.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |
| alsactl | root | 0 | 627 | 4.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 非常低 |
| atd | root | 0 | 1002 | 44.0 KiB | 不适用 | 不适用 | 不适用 | 不适用 | 普通 |

- 【操作要求 4】利用「系统监视器」查看所有的文件系统
- ① 在「系统监视器」窗口单击「文件系统」选项卡，显示当前 RHEL Server 5 系统中主要的文件系统。
- ② 单击「编辑」菜单中的「首选项」，弹出「系统监视器首选项」对话框。在「文件系统」选项卡，选中「显示全部文件系统」复选框，要求显示出全部的文件系统，最后单「关闭」按钮。

③ 「文件系统」选项卡显示全部的文件系统的信息



【操作要求 4】利用「系统日志」工具查看系统日志

超级用户依次单击「系统」菜单=>「管理」=>「系统日志」，打开「系统日志」窗口。可分别查看各类系统日志。

| | | 已更新 今天 00时00分21秒 | |
|--------------------------|-----------------|------------------|---|
| boot.log | Oct 4 00:51:56 | master | crond[1121]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 88% if used.) |
| btmtp | Oct 4 00:51:58 | master | crond[1121]: (CRON) INFO (running with inotify support) |
| | Oct 3 17:00:02 | master | CROND[3629]: (root) CMD (/usr/lib64/sa/sa1 1) |
| ▶ cron | Oct 3 17:01:01 | master | CROND[3684]: (root) CMD (run-parts /etc/cron.hourly) |
| | Oct 3 17:01:01 | master | run-parts(/etc/cron.hourly)[3684]: starting 0anacron |
| dmesg | Oct 3 17:01:01 | master | anacron[3701]: Anacron started on 2021-10-03 |
| ▶ firewallld | Oct 3 17:01:01 | master | anacron[3701]: Will run job 'cron.daily' in 7 min. |
| | Oct 3 17:01:01 | master | anacron[3701]: Will run job 'cron.weekly' in 27 min. |
| grubby-prune_debug | Oct 3 17:01:01 | master | anacron[3701]: Will run job 'cron.monthly' in 47 min. |
| ▶ maillog | Oct 3 17:01:01 | master | anacron[3701]: Jobs will be executed sequentially |
| ▶ messages | Oct 3 17:01:01 | master | run-parts(/etc/cron.hourly)[3703]: finished 0anacron |
| ▶ secure | Oct 3 17:01:01 | master | run-parts(/etc/cron.hourly)[3684]: starting mcelog.cron |
| | Oct 3 17:01:01 | master | run-parts(/etc/cron.hourly)[3709]: finished mcelog.cron |
| spooler | Oct 3 17:05:41 | master | crond[1121]: (CRON) INFO (Shutting down) |
| vmware-network.1.log | Oct 9 10:21:26 | master | crond[1034]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 20% if used.) |
| vmware-network.2.log | Oct 9 10:21:29 | master | crond[1034]: (CRON) INFO (running with inotify support) |
| vmware-network.3.log | Oct 9 10:30:01 | master | CROND[2767]: (root) CMD (/usr/lib64/sa/sa1 1) |
| vmware-network.4.log | Oct 14 09:53:06 | master | crond[1034]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 82% if used.) |
| vmware-network.log | Oct 14 09:53:07 | master | crond[1034]: (CRON) INFO (running with inotify support) |
| vmware-vgaauthsvc.log.0 | Oct 14 10:00:01 | master | CROND[2799]: (root) CMD (/usr/lib64/sa/sa1 1) |
| vmware-vmtoolsd-root.log | Oct 14 10:01:01 | master | CROND[2828]: (root) CMD (run-parts /etc/cron.hourly) |
| | Oct 14 10:01:01 | master | run-parts(/etc/cron.hourly)[2828]: starting 0anacron |
| | Oct 14 10:01:01 | master | anacron[2844]: Anacron started on 2021-10-14 |
| | Oct 14 10:01:01 | master | anacron[2844]: Will run job 'cron.daily' in 7 min. |
| | Oct 14 10:01:01 | master | anacron[2844]: Will run job 'cron.weekly' in 27 min. |
| | Oct 14 10:01:01 | master | anacron[2844]: Will run job 'cron.monthly' in 47 min. |
| | Oct 14 10:01:01 | master | anacron[2844]: Jobs will be executed sequentially |
| | Oct 14 10:01:01 | master | run-parts(/etc/cron.hourly)[2846]: finished 0anacron |
| wpa-supplciant.log | Oct 14 10:01:01 | master | run-parts(/etc/cron.hourly)[2828]: starting mcelog.cron |
| Xorg.0.log | Oct 14 10:01:01 | master | run-parts(/etc/cron.hourly)[2852]: finished mcelog.cron |
| Xorg.9.log | Oct 14 10:27:20 | master | anacron[2844]: Job 'cron.daily' started |
| yum.log | Oct 14 10:27:25 | master | run-parts(/etc/cron.daily)[2926]: starting logrotate |
| | Oct 14 10:27:25 | master | run-parts(/etc/cron.daily)[2943]: finished logrotate |
| | Oct 14 10:27:25 | master | run-parts(/etc/cron.daily)[2926]: starting man-db.cron |
| | Oct 14 10:27:42 | master | run-parts(/etc/cron.daily)[57800]: finished man-db.cron |

【实验总结】

本次实验我掌握了手工启动后台作业的方法，学会了一些与进程和作业管理相关的 Shell 命令；加深了对进程调度的理解，调度的方法有 at 调度、batch 调度、cron 调度，不同调度的使用方法、参数也略有差别；此外还学习了如何使用系统工具来进行性能的监视，使用系统监视器工具监视 CPU 的使用情况、查看当前的所有进程、查看所有的文件系统、查看系统日志等。

本次实验比较简单，按照实验指导书上的步骤完成即可，但由于更新等原因，少数命令的运行结果与指导书上的有差别，但通过查阅资料也都得到了解决。