

# 《计算机操作系统》实验报告

## 实验题目：Linux 进程调度与系统监视

姓名：严昕宇      学号：20121802      实验日期：2022.11.01

### 实验环境：

实验设备：Lenovo Thinkbook16+ 2022

操作系统：Ubuntu 22.04.1 LTS 64 位

### 实验目的：

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 cron 调度的设置方法
4. 了解进行系统性能监视的基本方法

### 实验内容：

1. 作业和进程的基本管理
2. at 进程调度
3. cron 进程调度
4. 系统性能监视

## 一、作业和进程的基本管理

### 操作过程 1：

【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

#### 【操作步骤】

- ① 以超级用户（root）身份登录到 RHEL Server 5 字符界面。
- ② 输入命令“vi f4”，在前台启动 vi 文本编辑器并打开 f4 文件。
- 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。
- 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。

### 结果 1：

由于 Ubuntu 系统中不存在 inittab 文件，此处改为 Public 文件夹

```
yanxinyu@Thinkbook16-2022:~$ su
密码:
root@Thinkbook16-2022:/home/yanxinyu# vi f4

[1]+  Stopped                  vi f4
root@Thinkbook16-2022:/home/yanxinyu# find / -name inittab >f5 &
[2] 6222
```

## 操作过程 2:

【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

【操作步骤】

- ① 输入命令“jobs”，查看当前系统中的所有作业。
- 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。
- 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”。
- ⑤ 输入命令“kill -9 %1”，终止“vi f4”作业。
- ⑥ 稍等片刻，输入命令“jobs”，查看当前没有任何作业。
- 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。
- ⑧ 输入命令“who -H”，查看用户信息。

## 结果 2:

```
root@Thinkbook16-2022:/home/yanxinyu#  
[2]-  Done                  find / -name Public > f5  
root@Thinkbook16-2022:/home/yanxinyu# jobs  
[1]+  Stopped                vi f4  
root@Thinkbook16-2022:/home/yanxinyu# cat f5  
/home/yanxinyu/snap/snapd-desktop-integration/14/Public  
root@Thinkbook16-2022:/home/yanxinyu# jobs  
[1]+  Stopped                vi f4  
root@Thinkbook16-2022:/home/yanxinyu# kill -9 %1  
[1]+  Killed                  vi f4  
root@Thinkbook16-2022:/home/yanxinyu# jobs  
root@Thinkbook16-2022:/home/yanxinyu# ps -l  
F S  UID      PID     PPID  C  PRI  NI ADDR SZ WCHAN  TTY          TIME CMD  
4 S   0       3359    3319  0   80   0  -   5525 do_wai pts/0    00:00:00 su  
4 S   0       3360    3359  0   80   0  -   4787 do_wai pts/0    00:00:00 bash  
4 R   0       3465    3360  0   80   0  -   5331 -      pts/0    00:00:00 ps  
root@Thinkbook16-2022:/home/yanxinyu# who -H  
NAME      LINE      TIME          COMMENT  
yanxinyu  tty2      2022-11-08 17:29 (tty2)  
root@Thinkbook16-2022:/home/yanxinyu#
```

## 二、at 进程调度

### 操作过程 1:

【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

【操作步骤】

- 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。
- 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。

### 结果 1:

```
root@Thinkbook16-2022:/home/yanxinyu# at 00:00 01012023
warning: commands will be executed using /bin/sh
at Sun Jan  1 00:00:00 2023
at> wall Happy New Year!
at> <EOT>
job 1 at Sun Jan  1 00:00:00 2023
```

### 操作过程 2:

【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

【操作步骤】

- 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。
- 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。

### 结果 2:

```
root@Thinkbook16-2022:/home/yanxinyu# at now +5 minutes
warning: commands will be executed using /bin/sh
at Tue Nov  8 17:50:00 2022
at> wall please logout; the computer will restart.
at> shutdown -r +2
at> <EOT>
job 2 at Tue Nov  8 17:50:00 2022
```

### 操作过程 3:

【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

【操作步骤】

方法一：

- 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。
- ② 输入“atrm1”命令删除作业号为 1 的 at 调度，并再次输入“atq”命令查看剩余的所有 at 调度内容。
- 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。

### 结果 3:

```
root@Thinkbook16-2022:/home/yanxinyu# atq
1      Sun Jan  1 00:00:00 2023 a root
4      Tue Nov  8 18:02:00 2022 a root
root@Thinkbook16-2022:/home/yanxinyu# atrm 1
root@Thinkbook16-2022:/home/yanxinyu# atq
4      Tue Nov  8 18:02:00 2022 a root
```



### 三、cron 进程调度

#### 操作过程 1:

【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

##### 【操作步骤】

- 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。
- 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”。
- 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”，
- ④ 保存并退出编辑器，显示“crontab: installing new crontab”信息。
- ⑤ 输入命令“crontab -l”，查看 helen 用户的 cron 调度内容。
- ⑥ 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。
- ⑦ 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。

#### 结果 1:

```
# 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
30 8 * * * ps >ps.log
-- INSERT --
```

```

crontab: installing new crontab
yanxinyu@Thinkbook16-2022:~$ crontab -l
# 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
30 8 * * * ps >ps.log

```

```

yanxinyu@Thinkbook16-2022:~$ su
密码:
root@Thinkbook16-2022:/home/yanxinyu# date 11080829
2022年 11月 08日 星期二 08:29:00 CST
root@Thinkbook16-2022:/home/yanxinyu# exit
exit
yanxinyu@Thinkbook16-2022:~$ cat ps.log

```

PID	TTY	TIME	CMD
1855	?	00:00:01	systemd
1856	?	00:00:00	(sd-pam)
1862	?	00:00:00	pipewire
1863	?	00:00:00	pipewire-media-
1864	?	00:00:01	pulseaudio
1865	?	00:00:01	snapd-desktop-i
1873	?	00:00:00	gnome-keyring-d
1878	?	00:00:01	dbus-daemon
1880	?	00:00:00	gvfsd
1889	?	00:00:00	gvfsd-fuse
1906	?	00:00:00	xdg-document-po
1909	?	00:00:00	xdg-permission-

## 操作过程 2:

【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

### 【操作步骤】

- 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”。
- 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。
- 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，那么说明新增加的 crontab 调度设置成功。

## 结果 2:

```
# 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
30 8 * * * ps >ps.log
0 0 * */3 * who >who.log
~
~
-- INSERT --
```



```
crontab: installing new crontab
yanxinyu@Thinkbook16-2022:~$ su
密码:
root@Thinkbook16-2022:/home/yanxinyu# date 03312359
2022年 03月 31日 星期四 23:59:00 CST
root@Thinkbook16-2022:/home/yanxinyu# exit
exit
yanxinyu@Thinkbook16-2022:~$ cat who.log
yanxinyu tty2          2022-11-08 18:20 (tty2)
```

## 操作过程 3:

【操作要求 3】查看 cron 调度内容，最后删除此调度

【操作步骤】

- ① 输入命令“crontab -l”，查看 cron 调度内容。
- 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。

## 结果 3:

```
yanxinyu@Thinkbook16-2022:~$ crontab -l
# 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
30 8 * * * ps >ps.log
0 0 * */3 * who >who.log
yanxinyu@Thinkbook16-2022:~$ crontab -r
yanxinyu@Thinkbook16-2022:~$ crontab -l
no crontab for yanxinyu
```



## 四、系统性能监视

### 操作过程 1:

【操作要求 1】利用 Shell 命令监视系统性能

【操作步骤】

- 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。
- ② 按下 M 键，所有进程按照内存使用率排列。
- ③ 按下 T 键，所有进程按照执行时间排列。
- ④ 最后按下 P 键，恢复按照 CPU 使用率排列所有进程。
- ⑤ 按下 CTRL+C 组合键结束 top 命令。

### 结果 1:

- 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 - 18:46:28 up 34 min, 1 user, load average: 0.00, 0.00, 0.06
任务: 296 total, 1 running, 295 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6.5 us, 2.0 sy, 0.0 ni, 91.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3889.8 total, 1207.9 free, 1239.4 used, 1442.5 buff/cache
MiB Swap: 2140.0 total, 2140.0 free, 0.0 used, 2319.4 avail Mem
```

进程号	USER	PR	NI	VIRT	RES	SHR	%CPU	%MEM	TIME+	COMMAND
2019	yanxinyu	20	0	4617628	292460	160680 S	10.2	7.3	0:36.55	gnome-shell
4185	yanxinyu	20	0	671904	76764	60320 S	6.6	1.9	0:07.47	gnome-terminal-
2550	yanxinyu	20	0	2877376	91272	73960 S	1.4	2.3	0:02.80	gjs
2206	yanxinyu	20	0	323948	12160	7336 S	0.8	0.3	0:03.23	ibus-daemon
650	systemd+	20	0	14824	6140	5344 S	0.6	0.2	0:03.25	systemd-oomd
2262	yanxinyu	20	0	357140	29096	18236 S	0.6	0.7	0:02.31	ibus-extension-
291	root	20	0	0	0	0 I	0.3	0.0	0:00.88	kworker/u256:28-events_freezable_power_
788	root	20	0	252396	8492	7156 S	0.3	0.2	0:04.90	vmtoolsd
2184	yanxinyu	20	0	458680	29332	21044 S	0.3	0.7	0:00.44	xdg-desktop-por
4791	root	20	0	0	0	0 I	0.3	0.0	0:00.07	kworker/u256:2-flush-8:0
4914	yanxinyu	20	0	21936	4248	3380 R	0.3	0.1	0:00.29	top
1	root	20	0	167788	13024	8108 S	0.0	0.3	0:06.29	systemd
2	root	20	0	0	0	0 S	0.0	0.0	0:00.04	kthreadd
3	root	0	-20	0	0	0 I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0 I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0 I	0.0	0.0	0:00.00	netns

- ② 按下 M 键，所有进程按照内存使用率排列。

```
top - 18:47:07 up 35 min, 1 user, load average: 0.00, 0.00, 0.06
任务: 296 total, 1 running, 295 sleeping, 0 stopped, 0 zombie
%Cpu(s): 7.5 us, 1.6 sy, 0.0 ni, 91.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3889.8 total, 1215.4 free, 1231.8 used, 1442.6 buff/cache
MiB Swap: 2140.0 total, 2140.0 free, 0.0 used, 2327.1 avail Mem
```

进程号	USER	PR	NI	VIRT	RES	SHR	%CPU	%MEM	TIME+	COMMAND
2019	yanxinyu	20	0	4617632	292396	160680 S	10.6	7.3	0:37.24	gnome-shell
2365	yanxinyu	20	0	1114524	111904	70688 S	0.0	2.8	0:04.96	snap-store
2550	yanxinyu	20	0	2877376	91268	73960 S	1.7	2.3	0:02.87	gjs
4185	yanxinyu	20	0	671904	76764	60320 S	8.0	1.9	0:07.89	gnome-terminal-
2278	yanxinyu	20	0	858260	71544	55556 S	0.0	1.8	0:00.52	evolution-alarm
2378	yanxinyu	20	0	210232	69244	46444 S	0.0	1.7	0:00.80	Xwayland
2556	yanxinyu	20	0	533188	69036	51756 S	0.0	1.7	0:00.35	gsd-xsettings
2686	yanxinyu	20	0	737376	59792	45524 S	0.0	1.5	0:00.53	gnome-calendar
2276	yanxinyu	20	0	304984	46224	30492 S	0.3	1.2	0:04.53	vmtoolsd
911	root	20	0	948548	42252	19816 S	0.0	1.1	0:05.47	snappd
2682	yanxinyu	20	0	402776	40844	26452 S	0.0	1.0	0:00.50	seahorse
2097	yanxinyu	20	0	570236	40432	33220 S	0.0	1.0	0:00.12	goa-daemon
3148	yanxinyu	20	0	541924	34348	25496 S	0.0	0.9	0:00.30	update-notifier
2162	yanxinyu	20	0	681588	33480	27700 S	0.0	0.8	0:00.42	evolution-addre
4918	yanxinyu	39	19	407860	31000	25272 S	0.0	0.8	0:00.04	tracker-extract
2642	root	20	0	401404	30964	25488 S	0.0	0.8	0:01.03	fwupd

- ③ 按下 T 键，所有进程按照执行时间排列。

top - 18:48:00 up 36 min, 1 user, load average: 0.00, 0.00, 0.05  
任务: 296 total, 1 running, 295 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 1.1 us, 0.6 sy, 0.0 ni, 98.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 3889.8 total, 1209.3 free, 1237.7 used, 1442.8 buff/cache  
MiB Swap: 2140.0 total, 2140.0 free, 0.0 used, 2321.2 avail Mem

进程号	USER	PR	NI	VIRT	RES	SHR	%CPU	%MEM	TIME+	COMMAND	
2019	yanxinyu	20	0	4617636	292360	160680	S	2.7	7.3	0:38.05	gnome-shell
4185	yanxinyu	20	0	671904	76764	60320	S	1.6	1.9	0:08.23	gnome-terminal-
1	root	20	0	167788	13024	8108	S	0.0	0.3	0:06.30	systemd
1301	root	20	0	455340	26804	19840	S	0.0	0.7	0:06.23	packagekitd
911	root	20	0	948548	42252	19816	S	0.0	1.1	0:05.47	snapp
708	root	20	0	252396	8492	7156	S	0.0	0.2	0:05.11	vmtoolsd
2365	yanxinyu	20	0	1114524	111904	70688	S	0.0	2.8	0:04.97	snap-store
2276	yanxinyu	20	0	304340	45768	30492	S	0.5	1.1	0:04.72	vmtoolsd
650	systemd+	20	0	14824	6140	5344	S	0.0	0.2	0:03.38	systemd-oomd
2206	yanxinyu	20	0	323948	12160	7336	S	0.0	0.3	0:03.35	ibus-daemon
2550	yanxinyu	20	0	2877376	91296	73960	S	0.0	2.3	0:02.93	gjs
2262	yanxinyu	20	0	357140	29096	18236	S	0.0	0.7	0:02.32	ibus-extension-
899	root	20	0	251660	10788	7304	S	0.0	0.3	0:02.15	polkitd
1864	yanxinyu	9	-11	1430880	26304	20780	S	0.0	0.7	0:01.70	pulseaudio
871	message+	20	0	11088	6716	4040	S	0.0	0.2	0:01.64	dbus-daemon
1855	yanxinyu	20	0	17936	10452	7856	S	0.0	0.3	0:01.59	systemd

- ④ 最后按下 P 键，恢复按照 CPU 使用率排列所有进程。

top - 18:48:31 up 36 min, 1 user, load average: 0.00, 0.00, 0.05

任务: 293 total, 1 running, 292 sleeping, 0 stopped, 0 zombie

%Cpu(s): 7.4 us, 1.4 sy, 0.0 ni, 91.2 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

MiB Mem : 3889.8 total, 1206.4 free, 1240.6 used, 1442.8 buff/cache

MiB Swap: 2140.0 total, 2140.0 free, 0.0 used, 2318.2 avail Mem

进程号	USER	PR	NI	VIRT	RES	SHR	%CPU	%MEM	TIME+	COMMAND	
2019	yanxinyu	20	0	4617640	292436	160680	S	11.0	7.3	0:38.63	gnome-shell
4185	yanxinyu	20	0	671904	76764	60320	S	7.6	1.9	0:08.58	gnome-terminal-
2550	yanxinyu	20	0	2877380	91260	73960	S	2.0	2.3	0:03.02	gjs
291	root	20	0	0	0	0	I	0.3	0.0	0:00.95	kworker/u256:28-events_unbound
408	root	-51	0	0	0	0	S	0.3	0.0	0:01.10	irq/16-vmwgfx
2206	yanxinyu	20	0	323948	12160	7336	S	0.3	0.3	0:03.42	ibus-daemon
2262	yanxinyu	20	0	357140	29096	18236	S	0.3	0.7	0:02.33	ibus-extension-
2276	yanxinyu	20	0	302528	43668	30492	S	0.3	1.1	0:04.87	vmtoolsd
4753	root	20	0	0	0	0	I	0.3	0.0	0:00.19	kworker/1:2-events
4914	yanxinyu	20	0	21936	4248	3380	R	0.3	0.1	0:00.84	top
4982	root	20	0	0	0	0	I	0.3	0.0	0:00.01	kworker/0:0-events
1	root	20	0	167788	13024	8108	S	0.0	0.3	0:06.30	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.04	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns

- ⑤ 按下 CTRL+C 组合键结束 top 命令。

## 操作过程 2：

【操作要求 2】利用「系统监视器」工具监视 CPU 使用情况

【操作步骤】

- 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。

结果 2:



操作过程 3:

【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

【操作步骤】

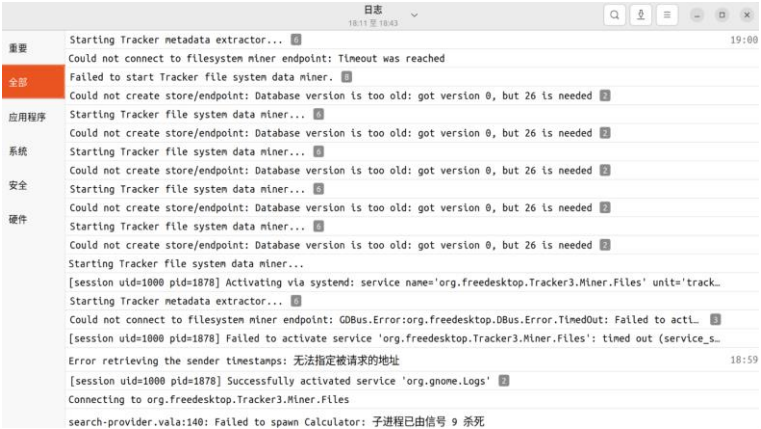
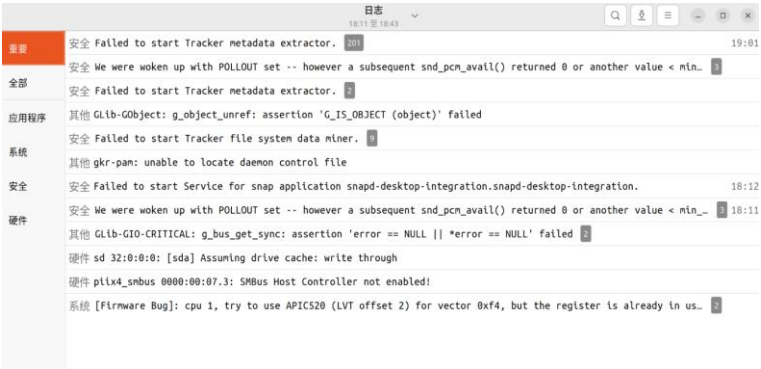
- ① 在「系统监视器」窗口单击「进程列表」选项卡，默认显示当前用户启动的所有进程。单击「查看」菜单，选中「所有的进程」单选按钮，并选中「依赖关系」复选框，则显示系统中所有的进程。
- ② 单击「编辑」菜单中的「首选项」，弹出「系统监视器首选项」对话框。在「进程」选项卡，选中「进程域」栏的「用户」复选框，要求显示出启动进程的用户。单击「关闭」按钮，显示进程的各种信息。

结果 3

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
gdm-wayland-session	yanxinyu	0.00	1918	479.2 kB	12.3 kB	不适用	不适用	不适用	普通
gnome-session-binary	yanxinyu	0.00	1931	1.7 MB	2.3 MB	不适用	不适用	不适用	普通
gnome-keyring-daemon	yanxinyu	0.00	1873	987.1 kB	532.5 kB	4.1 kB	不适用	不适用	普通
systemd	yanxinyu	0.00	1855	2.7 MB	18.2 MB	3.2 MB	不适用	不适用	普通
at-spi2-registrd	yanxinyu	0.00	2189	684.0 kB	不适用	不适用	不适用	不适用	普通
dbus-daemon	yanxinyu	0.00	1878	2.2 MB	不适用	不适用	不适用	不适用	普通
dconf-service	yanxinyu	0.00	2105	639.0 kB	77.8 kB	81.9 kB	不适用	不适用	普通
evolution-addressbook-fact	yanxinyu	0.00	2162	5.9 MB	3.1 MB	36.9 kB	不适用	不适用	普通
evolution-calendar-factory	yanxinyu	0.00	2112	4.0 MB	5.1 MB	不适用	不适用	不适用	普通
evolution-source-registrd	yanxinyu	0.00	2088	3.7 MB	3.4 MB	不适用	不适用	不适用	普通
gjs	yanxinyu	0.00	2190	4.9 MB	不适用	不适用	不适用	不适用	普通
gjs	yanxinyu	0.00	2454	5.0 MB	不适用	不适用	不适用	不适用	普通
gnome-calendar	yanxinyu	0.00	2686	14.6 MB	1.5 MB	不适用	不适用	不适用	普通
gnome-session-binary	yanxinyu	0.00	1995	3.0 MB	7.8 MB	4.1 kB	不适用	不适用	普通
at-spi-bus-launcher	yanxinyu	0.00	2015	700.4 kB	不适用	不适用	不适用	不适用	普通
dbus-daemon	yanxinyu	0.00	2027	483.3 kB	不适用	不适用	不适用	不适用	普通
evolution-alarm-notifier	yanxinyu	0.00	2278	16.4 MB	2.3 MB	不适用	不适用	不适用	普通
gsd-disk-utility-notifier	yanxinyu	0.00	2302	1.4 MB	24.6 kB	不适用	不适用	不适用	普通
update-notifier	yanxinyu	0.00	3148	9.1 MB	2.5 MB	14.4 MB	不适用	不适用	普通
gnome-session-ctl	yanxinyu	0.00	1982	442.4 kB	20.5 kB	不适用	不适用	不适用	普通



# 结果 5



## 实验体会

本次实验以进程操作为主，让我熟悉了许多与进程相关的操作。了解到可以交给操作系统指定时间的任务。与此同时，使用 `top` 指令让我直观看到了系统统计信息。`top` 命令是 Linux 下常用的性能分析工具，能够实时监控系统的运行状态，主要显示系统中各个进程的资源使用情况，并根据 CPU、内存等参数进行排序，类似于 Windows 系统中的任务管理器，可以实时查看系统的各种资源状态，是测试工程师尤其是性能测试工程师必须要掌握的命令。

结合理论课所学，我逐渐能够意识到，操作系统是一个极其庞大复杂却精密的体系，底层有许多我们不可见而又被精心设计过的巧妙构思，这都需要我在实践中不断学习和深入。