

STAT243 Lecture 3.5 Bash Shell-Managing Processes

1 What Is a Process

- **定义：**
进程（process）是正在被执行的程序实例。
- **进程属性**包括：
 - 生命周期（lifetime）
 - 进程 ID（PID）
 - 用户 ID（UID）
 - 组 ID（GID）
 - 父进程 ID（PPID）
 - 环境变量（environment）
 - 当前工作目录（current working directory）
- 每当执行一个命令或程序时，系统都会启动一个或多个进程来执行任务。

2 Monitoring

2.1 Monitoring Processes

2.1.1 Using `ps`

- **基本命令：**查看当前 shell 的子进程：

	Shell
1	\$ <code>ps</code>
2	PID TTY TIME CMD
3	19370 pts/3 00:00:00 bash
4	22846 pts/3 00:00:00 ps

- **详细显示当前 shell 的进程层级：**

	Shell
1	\$ <code>ps -f</code>
2	UID PID PPID C STIME TTY TIME CMD
3	jarrod 19370 19368 0 10:51 pts/3 00:00:00 bash
4	jarrod 22850 19370 0 14:57 pts/3 00:00:00 ps -f

- **显示系统中所有进程：**

	Shell
1	\$ <code>ps -ef</code>
2	UID PID PPID C STIME TTY TIME CMD
3	root 1 0 0 Aug21 ? 00:00:05 /usr/lib/systemd
4	root 2 0 0 Aug21 ? 00:00:00 [kthreadd]
5	root 3 2 0 Aug21 ? 00:00:07 [ksoftirqd/0]
6	...
7	jarrod 16219 16210 0 07:19 tty1 00:00:00 -bash
8	jarrod 16361 16219 0 07:19 tty1 00:00:00 /bin/sh /bin/startx

- **常用选项：**
 - `-u`：显示每个进程的 CPU 和内存占用百分比。

- **-o**：自定义输出字段，例如：

```
Shell
1 $ ps -o pid,ni,pcpu,pmem,user,comm
2 PID    NI %CPU %MEM  USER    COMMAND
3 18124   0  0.0  0.0  jarrod   bash
4 22963   0  0.0  0.0  jarrod   ps
```

- 查看进程树结构：

```
Shell
1 $ pstree
```

显示父进程与子进程之间的层级关系。

2.1.2 Using **top**

- 动态查看系统进程状态：

```
Shell
1 $ top
```

输出示例：

```
1 top - 13:49:07 up 1:49, 3 users, load average: 0.10, 0.15, 0.18
2 Tasks: 160 total, 1 running, 158 sleeping, 1 stopped, 0 zombie
3 %Cpu(s): 2.5 us, 0.5 sy, 0.0 ni, 96.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
4 KiB Mem : 7893644 total, 5951552 free, 1085584 used, 856508 buff/cache
5 KiB Swap: 7897084 total, 7897084 free, 0 used. 6561548 avail Mem
6
7 PID USER      PR  NI   VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
8 1607 jarrod    20   0 2333568 974888 212944 S   12.5  12.4   11:10.67 firefox
9 3366 jarrod    20   0 159828   4312   3624 R    6.2   0.1    0:00.01 top
```

- **RES**：进程实际使用的物理内存（Resident memory）。
- **%MEM**：占物理内存比例。
- **%CPU**：占用 CPU 核心比例（多线程进程可 >100%）。
- **TIME+**：进程累计运行时间。
- 按 **q** 退出。

Logic

在 **top** 中可以动态管理进程：

- 按 **r** 可调整进程优先级（renice）；
- 按 **k** 可终止进程（kill）。

2.2 Monitoring Memory Use

- 内存使用异常的典型表现：
 - 进程使用接近 **100% 内存**，但 **CPU 使用率较低**。
 - 表明程序正在使用磁盘作为虚拟内存（swap/paging），性能极差。
 - 此时程序大部分时间在磁盘读写，几乎无法完成任务。

Remark

如果程序陷入“paging”状态，可能长时间无响应。
通常应减少内存占用或使用分布式处理框架。

- 查看整机可用内存：

>_

Shell

1

\$ free -h

2

total

used

free

shared

buff/cache

available

3

Mem:

251G

998M

221G

2.6G

29G

247G

4

Swap:

7.6G

210M

7.4G

- 关注 **Mem** 行；
- 最关键的列是：
 - total**：总内存
 - used**：被进程实际占用的内存
 - available**：可立即使用的内存

Logic

free 列往往具有误导性，它不代表真实“可用”内存。
应始终参考 **available** 列来判断系统是否内存不足。