

# 进程

2023年4月24日 9:29

正在执行的程序，动态的概念。同时也是操作系统资源分配的基本单位。

虚拟内存的特点：

每一个进程独占一个内存空间（进程地址空间）

0~MAX所有的地址（虚拟地址）都是可用的（连续的空间）

不同进程的地址空间是彼此隔离的

# CPU资源

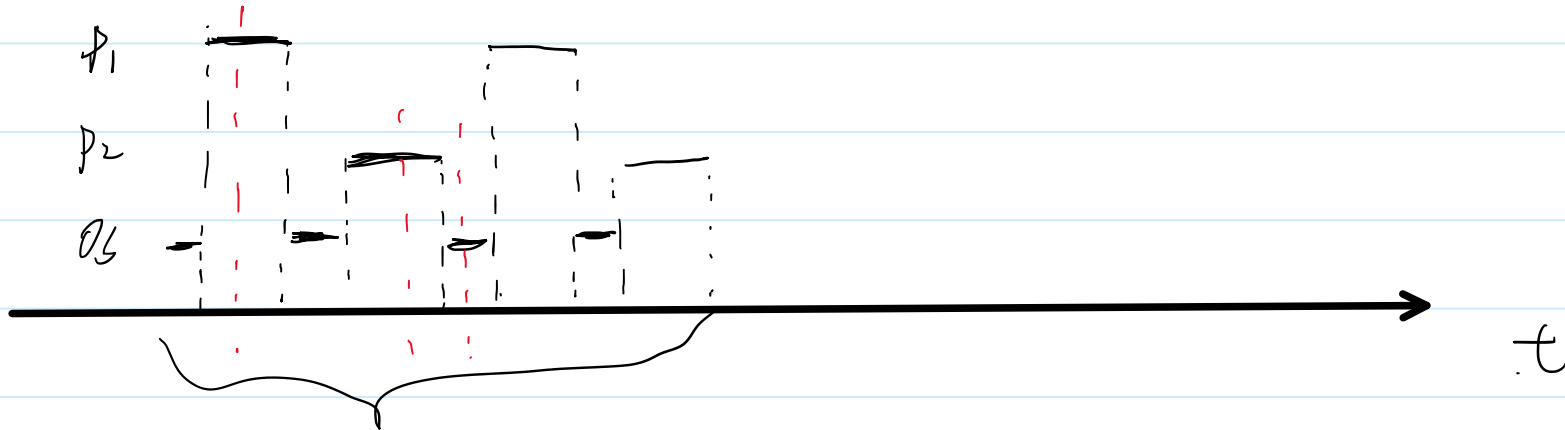
2023年4月24日

9:43

虚拟CPU：让每个进程以为自己CPU的独占者。

并发：某一段时间 多个进程同时运行

并行：某一个时刻 多个进程同时运行



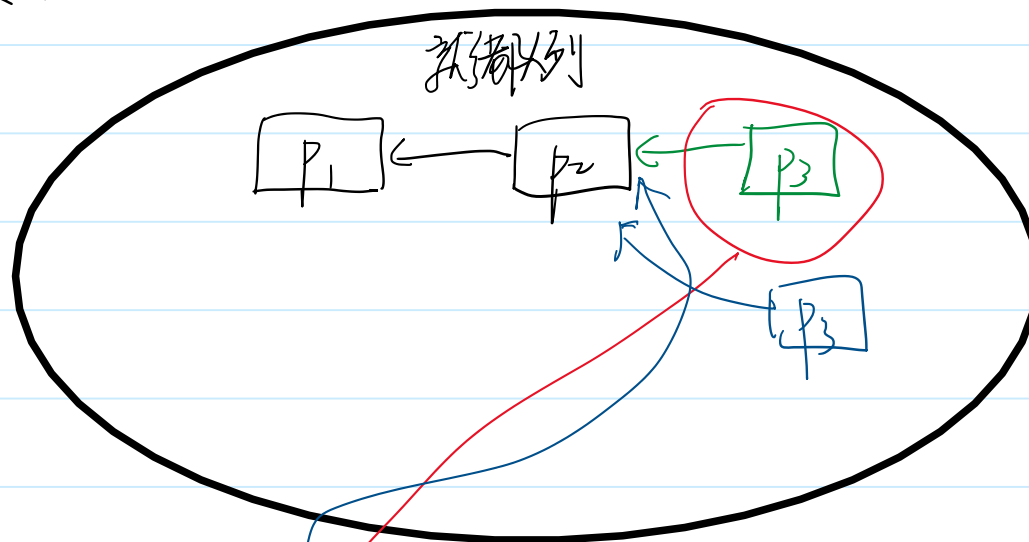
# 进程的调度

2023年4月24日 9:55

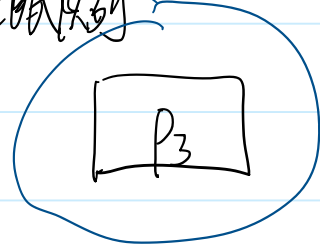
时间片轮转法

→ CFS 完全公平调度

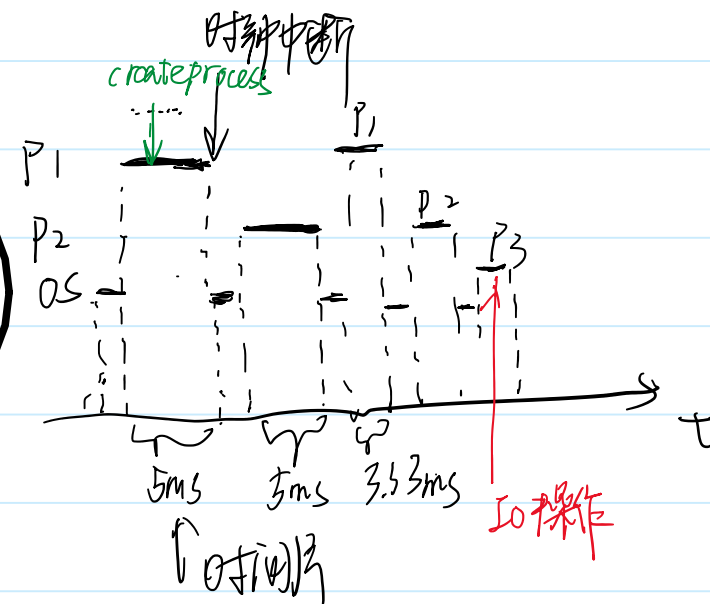
调度周期 10ms



睡眠队列



IO完成



就绪 ←→ 运行

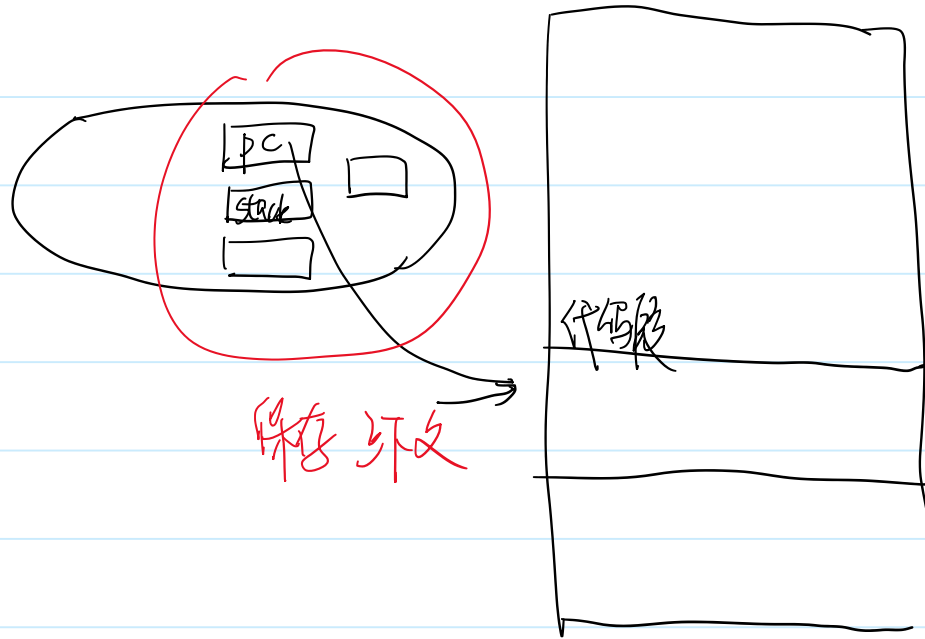
IO完成触发中断

阻塞

执行IO操作指令

# 进程的切换

2023年4月24日 10:19



进程的恢复

过程 → 上下文切换时保存

过程 → 恢复

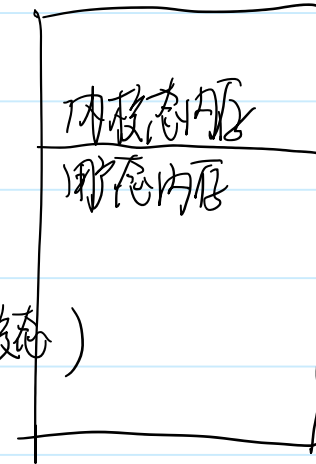
# 一些概念

2023年4月24日 10:25

内核态和用户态。

硬件cpu的特权级别：

内核态	所有指令	①外部硬件 ②访问内存(内核态)
用户态	部分指令	



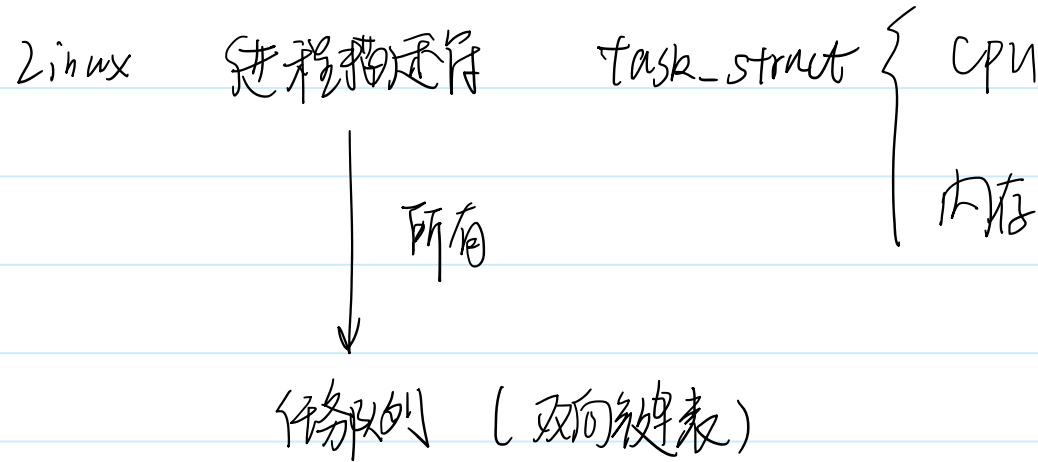
中断：硬件层面的事件机制      暂停原执行流， $\rightarrow$  切换内核态  $\rightarrow$  处理中断  $\rightarrow$  恢复原执行流

系统调用：无触发中断。在处理中断中，实现逻辑。

异常 int 0x80

# 进程控制块 PCB

2023年4月24日 11:07



```
struct mm_struct *mm;  
struct mm_struct *active_mm;  
  
/* Filesystem information: */  
struct fs_struct *fs;  
  
/* Open file information: */  
struct files_struct *files;
```

进程描述符 是 OS 管理进程所使用的数据结构

# pid

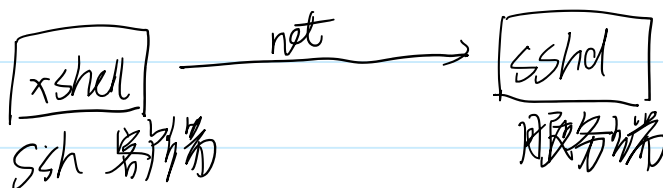
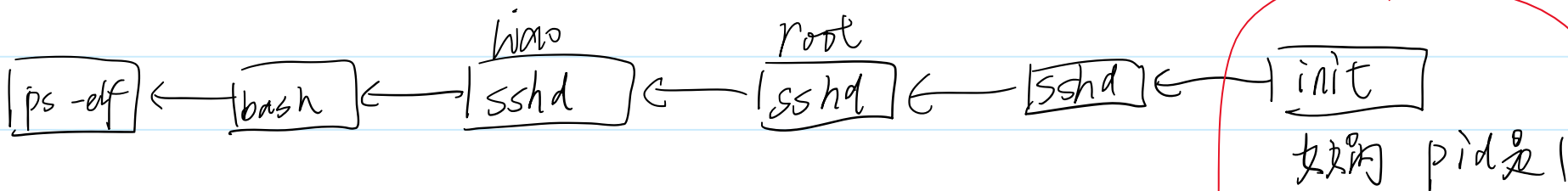
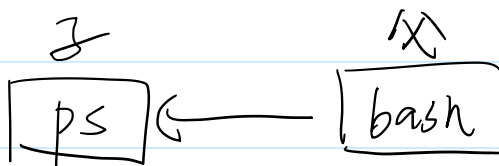
2023年4月24日 11:29

进程的标识 是一个非负整数: 那怎么某个进程

```
[liao@ubuntu 2 Linux]$ ps -l
```

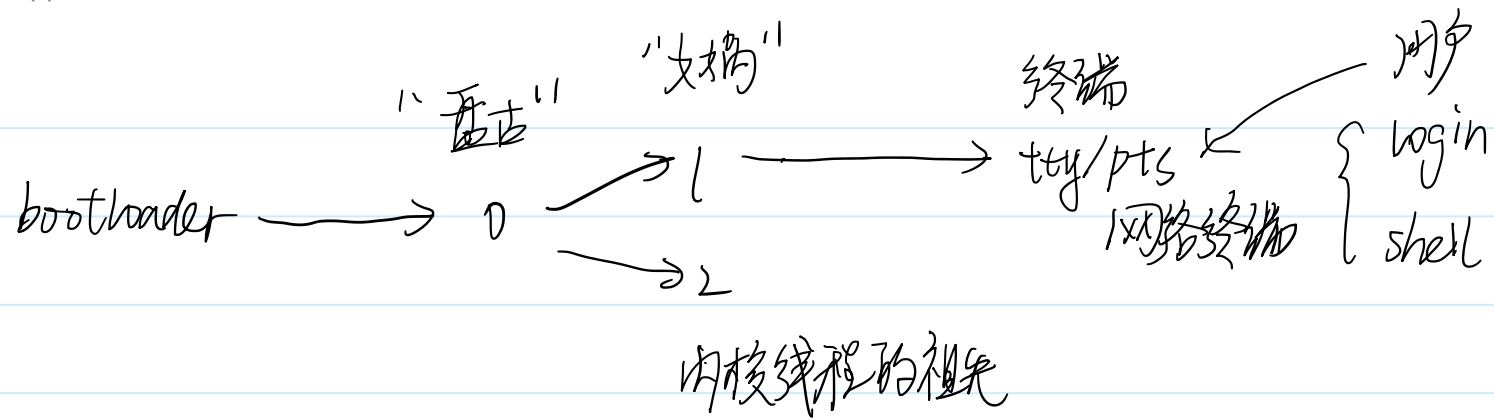
F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	14969	14968	0	80	0	-	5166	do_wai	pts/4	00:00:00	bash
0	R	1000	24455	14969	0	80	0	-	5012	-	pts/4	00:00:00	ps

bash是 shell的一种.



# 操作系统开机的过程

2023年4月24日 11:42



0 idle 空闲进程



# getpid

2023年4月24日

14:39

```
pid_t getpid(void);  
pid_t getppid(void);
```

```
#include <unistd.h>  
int main()  
{  
    printf("pid = %d, ppid = %d\n", getpid(), getppid());  
    return 0;  
}
```

```
[liao@ubuntu Linuxday 10]$ ./0_getpid  
pid = 24803, ppid = 14969  
[liao@ubuntu Linuxday 10]$ ./0_getpid  
pid = 24804, ppid = 14969
```

# getuid

2023年4月24日

14:46

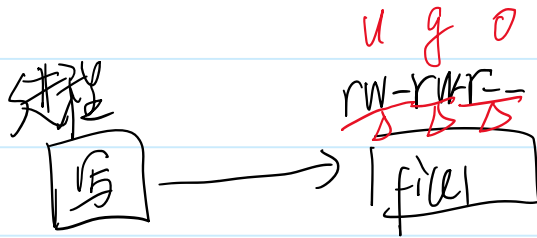
```
uid_t getuid(void);  
uid_t geteuid(void);
```

区分

## DESCRIPTION

`getuid()` returns the real user ID of the calling process.

`geteuid()` returns the effective user ID of the calling process.



# 进程的真实身份 ← 启动者.

2023年4月24日 14:55

liao 1000

```
-rwxrwxr-x 1 liao liao 19584 Apr 24 14:55 1_getuid*
```

test 1002

```
[liao@ubuntu Linuxday 10]$ ./1_getuid  
uid = 1000, euid = 1000
```

test 1002

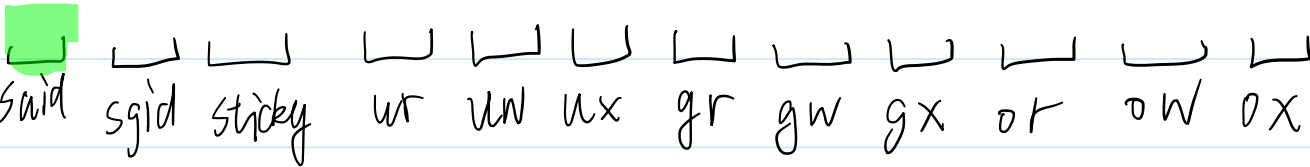
```
test@ubuntu:/home/liao/49code/2_Linux/Linuxday_10$ ./1_getuid  
test@ubuntu:/home/liao/49code/2_Linux/Linuxday_10$ ./1_getuid  
test@ubuntu:/home/liao/49code/2_Linux/Linuxday_10$ ./1_getuid  
uid = 1002, euid = 1002
```

```
int main()  
{  
    printf("uid = %d, euid = %d\n", getuid(), geteuid());  
    int fd = open("file1", O_RDWR);  
    ERROR_CHECK(fd, -1, "open");  
    ssize_t sret = write(fd, "hello", 5);  
    ERROR_CHECK(sret, -1, "write");  
    close(fd);  
    return 0;  
}
```

```
test@ubuntu:/home/liao/49code/2_Linux/Linuxday_10$ ./1_getuid  
uid = 1002, euid = 1002  
open: Permission denied
```

# 更改进程的有效身份

2023年4月24日 15:03

suid 权限. 

对于一个可执行程序文件，同时拥有 suid、u的x、o的x，通过该程序启动进程。

这个进程的有效身份  
是可执行程序文件的拥有者

```
[liao@ubuntu Linuxday 10]$ chmod u+s 1_getuid
[liao@ubuntu Linuxday 10]$ ll
total 64
drwxrwxr-x 2 liao liao 4096 Apr 24 14:59 ./
drwxrwxr-x 7 liao liao 4096 Apr 24 09:30 ../
-rwxrwxr-x 1 liao liao 19584 Apr 24 14:43 0_getpid*
-rw-rw-r-- 1 liao liao 108 Apr 24 14:43 0_getpid.c
-rwsrwxr-x 1 liao liao 19880 Apr 24 14:59 1_getuid*
```

```
test@ubuntu:/home/liao/49code/2_Linux/Linuxday_10$ ./1_getuid
uid = 1002, euid = 1000
```

```
[liao@ubuntu Linuxday 10]$ ll /usr/bin/sudo
-rwsr-xr-x 1 root root 166056 Apr 4 19:56 /usr/bin/sudo*
```

# 用户登录的原理

2023年4月24日 15:12

```
[liao@ubuntu Linuxday 10]$ ll /etc/shadow
-rw-r----- 1 root shadow 1997 Apr 12 15:30 /etc/shadow
```

明文.

salt

密文.

```
test:$6$SxIxUNPLMPUaGNdK$ZwVT2oVsFfdGOSILRTRDNRU0qockhpwGKgcdba7TRsMfz/UIoI1hhH3T0Jfsd24Abo6U9HtpDG6G.GbA4V4sE.:1
test1:$6$vBNU7.MSoZxMmqFg$CEAlCsFS6J0MDp.XanCVgKy8YyUMzph3NG7H5mMKLSJJSSff1JRwuAPgdx6d43sRwnCygZdd1V/9EqNIka2bs1:
```

加密函数  $\text{encrypt}(\text{password}, \text{salt}) \longrightarrow$  密文密码.

注册 password  $\xrightarrow{\text{生成}}$  salt + 密文

登录 password  $\xrightarrow{\text{找列}}$  salt  $\longrightarrow$  新密文.

```
[liao@ubuntu Linuxday 10]$ which passwd
/usr/bin/passwd
[liao@ubuntu Linuxday 10]$ ll /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 Nov 29 19:53 /usr/bin/passwd*
```

# getgid

2023年4月24日 15:30

```
gid_t getgid(void);  
gid_t getegid(void);
```

*sgid*

```
[liao@ubuntu Linuxday 10]$ chmod g+s 2_getgid  
[liao@ubuntu Linuxday 10]$ ll  
total 88  
drwxrwxr-x 2 liao liao 4096 Apr 24 16:00 ./  
drwxrwxr-x 7 liao liao 4096 Apr 24 09:30 ../  
-rwxrwxr-x 1 liao liao 19584 Apr 24 14:43 0_getpid*  
-rw-rw-r-- 1 liao liao 108 Apr 24 14:43 0_getpid.c  
-rwsrwxr-x 1 liao liao 19880 Apr 24 14:59 1_getuid*  
-rw-rw-r-- 1 liao liao 263 Apr 24 14:59 1_getuid.c  
-rwxrwsr-x 1 liao liao 19584 Apr 24 16:00 2_getgid*
```

```
-rw-rw-r-- 1 liao liao 27 Apr 24 14:42 Makefile  
[liao@ubuntu Linuxday 10]$ su test  
Password:  
test@ubuntu:/home/liao/49code/2_Linux/Linuxday_10$ ./2_getgid  
gid = 1002, egid = 1000
```

# sticky 粘滞位

2023年4月24日 16:03

目录文件 o+w

```
[liao@ubuntu Linuxday_10]$ chmod o+w dir1
[liao@ubuntu Linuxday_10]$ cd dir1/
[liao@ubuntu dir1]$ ll
total 8
drwxrwxrwx 2 liao liao 4096 Apr 24 16:04 ./
```

```
-rw-rw-r-- 1 test test 0 Apr 24 16:06 testfile2
test1@ubuntu:/home/liao/49code/2_Linux/Linuxday_10/dir1$ rm testfile2
rm: remove write-protected regular empty file 'testfile2'? y
test1@ubuntu:/home/liao/49code/2_Linux/Linuxday_10/dir1$ ll
total 8
drwxrwxrwx 2 liao liao 4096 Apr 24 16:07 ./
drwxrwxr-x 3 liao liao 4096 Apr 24 16:04 ../
```

← 可以删除其他那的文件，  
限制这个操作

## 同时拥有 o 的w和t

2023年4月24日 16:10

```
[liao@ubuntu dir1]$ cd ..  
[liao@ubuntu Linuxday_10]$ chmod o+t dir1/  
drwxrwxrwt 2 liao liao 4096 Apr 24 16:07 dir1/
```

```
test1@ubuntu:/home/liao/49code/2_Linux/Linuxday_10/dir1$ rm testfile2  
rm: remove write-protected regular empty file 'testfile2'? y  
rm: cannot remove 'testfile2': Operation not permitted
```

其他用户 可以新建文件，可以删除自己的文件，不可以删除别人的文件，



# 进程相关的命令

2023年4月24日 16:16

## 5.7 进程相关的命令

命令名	详细信息
ps	查看系统当中的进程
top	动态显示系统当中的进程
nice	用于shell脚本中，指定程序的优先级
renice	改变正在运行进程的优先级
kill	发送信号（可以给后台进程发送）
crontab	控制cron后台进程
bg	将暂停的进程放到后台

# ps -elf

2023年4月24日 16:17

[liao@ubuntu dir1]\$ ps -elf

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	STIME	TTY	TIME	CMD
4	S	root	1	0	0	80	0	- 25707	-		Apr20 ?		00:00:07	/sbin/init auto noprompt
1	S	root	2	0	0	80	0	-	0	-	Apr20 ?		00:00:00	[kthreadd] ← 内核线程
1	I	root	3	2	0	60	-20	-	0	-	Apr20 ?		00:00:00	[rcu_gp]
1	I	root	4	2	0	60	-20	-	0	-	Apr20 ?		00:00:00	[rcu_par_gp]
1	I	root	5	2	0	60	-20	-	0	-	Apr20 ?		00:00:00	[slub_flushwq]
1	I	root	6	2	0	60	-20	-	0	-	Apr20 ?		00:00:00	[netns]

F flag

S 状态

- D uninterruptible sleep (usually IO) ← 不可中断睡眠 (磁盘)
- I Idle kernel thread ← 空闲
- R running or runnable (on run queue) ← 就绪队列
- S interruptible sleep (waiting for an event to complete) ← 可中断睡眠 (设备、网络、管道...)
- T stopped by job control signal ← Ctrl+Z
- t stopped by debugger during the tracing ← gdb
- W paging (not valid since the 2.6.xx kernel)
- X dead (should never be seen)
- Z defunct ("zombie") process, terminated but not reaped by its parent ← 僵尸

C cpu使用率

已经也 资源未回收

# ps aux

2023年4月24日 16:17

```
[liao@ubuntu ~]$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.1	102828	11584	?	Ss	Apr20	0:07	/sbin/init auto noprompt
root	2	0.0	0.0	0	0	?	S	Apr20	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	Apr20	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	Apr20	0:00	[rcu_par_gp]
root	5	0.0	0.0	0	0	?	I<	Apr20	0:00	[slub_flushwq]
root	6	0.0	0.0	0	0	?	I<	Apr20	0:00	[netns]
root	8	0.0	0.0	0	0	?	I<	Apr20	0:00	[kworker/0:0H-events_highpri]
root	10	0.0	0.0	0	0	?	I<	Apr20	0:00	[mm_percpu_wq]
root	11	0.0	0.0	0	0	?	S	Apr20	0:00	[rcu_tasks_rude_]
root	12	0.0	0.0	0	0	?	S	Apr20	0:00	[rcu_tasks_trace]

**vsz** **VSZ** virtual memory size of the process in KiB (1024-byte units).

resident set size, the non-swapped physical memory that a task has used (in kilobytes).

< high-priority (not nice to other users)

N low-priority (nice to other users)

L has pages locked into memory (for real-time and custom IO)

s is a session leader

l is multi-threaded (using CLONE\_THREAD, like NPTL pthreads do)

+ is in the foreground process group

# free

2023年4月24日 16:48

free - Display amount of free and used memory in the system

```
[liao@ubuntu ~]$ free -m
```

	total	used	free	shared	buff/cache	available
Mem:	7915	1098	4822	7	1994	6527
Swap:	2047	0	2047			

buffers

缓冲

Memory used by kernel buffers (Buffers in /proc/meminfo)

先进先出

cache Memory used by the page cache and slabs (Cached and SReclaimable in /proc/meminfo)

缓存

在高建设设备上 复制一份低建设设备数据

# top

2023年4月24日 17:18

```
top - 17:19:36 up 4 days, 7:27, 5 users, load average: 0.07, 0.06, 0.01
Tasks: 287 total, 1 running, 286 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7915.8 total, 4821.0 free, 1099.4 used, 1995.4 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 6527.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
371	root	20	0	0	0	0	I	0.3	0.0	4:26.33	kworker/0:3-mpt_poll_0
1014	mysql	20	0	1305340	394496	37676	S	0.3	4.9	46:53.59	mysqld
1	root	20	0	102828	11584	8332	S	0.0	0.1	0:07.81	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	slub_flushwq

```
-----
us, user      : time running un-niced user processes
sy, system    : time running kernel processes
ni, nice      : time running niced user processes
id, idle      : time spent in the kernel idle handler
wa, IO-wait   : time waiting for I/O completion
hi            : time spent servicing hardware interrupts
si            : time spent servicing software interrupts
st            : time stolen from this vm by the hypervisor
```

# 优先级系统

2023年4月24日 17:33

两种 { 高优先级 实时  
          { 低 普通 CFS       $\rightarrow$  [60, 99]

SCHED\_FIFO: First in-first out scheduling  
SCHED\_RR: Round-robin scheduling } [-40, 59]

140个级别      ubuntu [-40, 99]      数值越大, 优先级越低.

nice值, 用来间接控制优先级.

NICE	PRI
0	80
20	60
19	99

NICE [-20, 19]

$\hookrightarrow$  用户能直接改

# nice 以某个nice值去启动进程

2023年4月24日 17:41

```
10 80  
[liao@ubuntu Linuxday 10]$ nice -n 10 ./3_while1  
^C  
19 99  
[liao@ubuntu Linuxday 10]$ nice -n 20 ./3_while1  
^C
```

} - 低优先.

```
[liao@ubuntu Linuxday 10]$ nice -n -10 ./3_while1  
nice: cannot set niceness: Permission denied  
^C  
[liao@ubuntu Linuxday 10]$ sudo nice -n -10 ./3_while1  
[sudo] password for liao:  
-10 70
```

```
^C  
[liao@ubuntu Linuxday 10]$ sudo nice -n -30 ./3_while1  
^C  
-20 60
```

不能通过修改nice值改策略

# renice 更改已经运行的进程的nice值

2023年4月24日 17:46

```
[liao@ubuntu Linuxday 10]$ ps -elf|grep 3_while1
0 R liao      26130    25630 99  80   0 -   591 -      17:46 pts/2    00:00:02 ./3_while1
0 S liao      26132      1796  0  80   0 -  4418 pipe_r 17:46 pts/0    00:00:00 grep --color
[liao@ubuntu Linuxday 10]$ renice -n 19 -p 26130
26130 (process ID) old priority 0, new priority 19
[liao@ubuntu Linuxday 10]$ ps -elf|grep 3_while1
0 R liao      26130    25630 99  99  19 -   591 -      17:46 pts/2    00:00:34 ./3_while1
```

```
[liao@ubuntu Linuxday 10]$ renice -n 10 -p 26130
renice: failed to set priority for 26130 (process ID): Permission denied
[liao@ubuntu Linuxday 10]$ sudo renice -n 10 -p 26130
[sudo] password for liao:
26130 (process ID) old priority 19, new priority 10
[liao@ubuntu Linuxday 10]$ ps -elf|grep 3_while1
0 R liao      26130    25630 99  90  10 -   591 -      17:46 pts/2    00:01:15 ./3_while1
0 S liao      26140      1796  0  80   0 -  4418 pipe_r 17:48 pts/0    00:00:00 grep --color=auto 3_while1
```