

- 系统调用的传参方式, 是寄存器传参
- wait和ready是有queue的, 新进程在哪个队列要看资源是否分配完全
- **system_call()函数实现了系统调用中断处理程序:**
 - 1.它首先把系统调用号和该异常处理程序用到的所有CPU寄存器保存到相应的栈中, SAVE_ALL
 - 2.把当前进程task_struct (thread_info) 结构的地址存放在ebx中
 - 3.对用户态进程传递来的系统调用号进行有效性检查。若调用号大于或等于NR_syscalls, 系统调用处理程序终止 (sys_call_table)
 - 4.若系统调用号无效, 函数就把-ENOSYS值存放在栈中eax寄存器所在的单元, 再跳到ret_from_sys_call()
 - 5.根据eax中所包含的系统调用号调用对应的特定服务例程

- if(fork() == 0) A else B-父亲执行B(>0), 儿子执行A(=0)

一个多线程进程fork出来的进程是多线程还是单线程的? 结论: 是单线程的。多线程进程调用fork () 时, 仅会将发起调用的线程复制到子进程中。子进程中该线程的线程ID与父进程中发起fork () 调用线程的线程ID相一致, 进程号不一致。

任一线程调用了exec () 系列函数之一时, 调用程序将被完全替换。除了调用exec () 的线程之外, 其他所有线程都将立即消失。调用exec () 之后, 调用线程的线程ID是不确定的。

```
gcc a.cpp -o a -w -lpthread
```

- 线程也有五种状态, 也有wait (blocked) 态。线程栈是私有的, 但是不享有自己的独立地址空间

1. 新建状态(New): 线程对象被创建后, 就进入了新建状态。例如, Thread thread = new Thread()。

2. 就绪状态(Runnable): 也被称为“可执行状态”。线程对象被创建后, 其它线程调用了该对象的start()方法, 从而来启动该线程。例如, thread.start()。处于就绪状态的线程, 随时可能被CPU调度执行。

3. 运行状态(Running): 线程获取CPU权限进行执行。需要注意的是, 线程只能从就绪状态进入到运行状态。

4. 阻塞状态(Blocked): 阻塞状态是线程因为某种原因放弃CPU使用权, 暂时停止运行。直到线程进入就绪状态, 才有机会转到运行状态。阻塞的情况分三种:

- (01) 等待阻塞 – 通过调用线程的wait()方法, 让线程等待某工作的完成。
- (02) 同步阻塞 – 线程在获取synchronized同步锁失败(因为锁被其它线程所占用), 它会进入同步阻塞状态。
- (03) 其他阻塞 – 通过调用线程的sleep()或join()或发出了I/O请求时, 线程会进入到阻塞状态。当sleep()状态超时、join()等待线程终止或者超时、或者I/O处理完毕时, 线程重新转入就绪状态。

5. 死亡状态(Dead): 线程执行完了或者因异常退出了run()方法, 该线程结束 **生命周期**。

- $EAT = (t_1 + t_2) * \alpha + (t_1 + t_2 + t_2) * \alpha$, TLB不命中的一次时间是去查页表的
- "Walk page table" 是一种用于查找虚拟内存地址映射到物理内存地址的方法。
- swap空间有两种形式:

Swap-space can be carried out in two forms:

- in the **normal file system**
e.g. **Windows** family
- in a **separate disk partition**
e.g. **Linux**、**Unix**、solaris

pagefile.sys文件

SWAP分区

- 启动文件一般指系统启动或用户登录等所执行配置文件；如系统配置 `/etc/profile`，用户配置 `~/.profile`
- 在Linux中终止一个进程有两种方式，如果是前台进程可以使用Ctrl+C键进行终止；如果是后台进程，那么需要使用kill命令来终止 `kill[参数][进程号]` `kill -9 21121`
- 主分区1~4，必须预留，所以你的拓展分区应该从hdb5开始

IDE1的第1个硬盘 (master) 的第1个主分区 `/dev/hda1`

IDE1的第1个硬盘 (master) 的第2个主分区 `/dev/hda2`

IDE1的第1个硬盘 (master) 的第3个主分区 `/dev/hda3`

IDE1的第1个硬盘 (master) 的第4个主分区 `/dev/hda4`

IDE1的第1个硬盘 (master) 的第1个逻辑分区 `/dev/hda5`

IDE1的第1个硬盘 (master) 的第2个逻辑分区 `/dev/hda6`

.....

IDE1的第2个硬盘 (slave) 的第1个主分区 `/dev/hdb1`

IDE1的第2个硬盘 (slave) 的第2个主分区 `/dev/hdb2`

.....

SCSI的第1个硬盘的第1个主分区 `/dev/sda1`

SCSI的第1个硬盘的第2个主分区 `/dev/sda2`

将 `/dev/hda1` 用唯读模式挂在 `/mnt` 之下。 `#mount -o ro /dev/hda1 /mnt`

- RAID (Redundant Array of Independent Disks)
- `find . -type f -name "*.png" (1)` find命令是根据**文件的属性**进行查找，如文件名，文件大小，所有者，所属组，是否为空，访问时间，修改时间等。
(2)grep是根据**文件的内容**进行查找，会对文件的每一行按照给定的模式(patter)进行匹配查找。
- 交换技术调入 / 调出整个进程，因此一个进程的大小要受内存容量大小的限制；而虚存中使用的调入 / 调出技术在内存和外存之间来回传递的是页面或分段，而不是整个进程，从而使得进程的地址映射具有更大的灵活性；覆盖程序段的最大长度要受内存容量大小的限制，而虚拟存储器中程序的最大长度不受内存容量的限制