**Linux中PCB的实现,以及其中所含的子结构体之间的关系.请用图形形式实现并辅助文字说明.**

**PCB的基础概念**:存放进程的管理和控制信息的数据结构,进程控制块是进程存在的**唯一标志**,系统运行中有若干个程序的PCB,它们常驻内存的PCB区.

采用的数据结构:PCB结构体,PCB链表或队列.

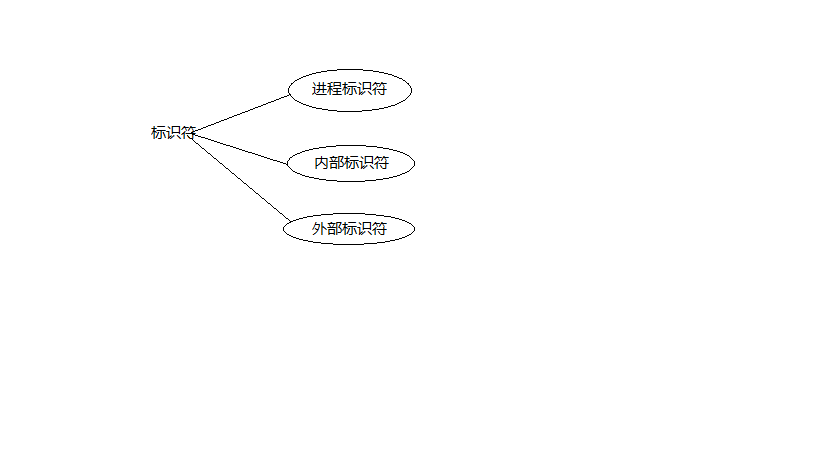
PCB的管理:将相同状态的(就绪,阻塞,空白)PCB链接成一个队列

索引方式:将形成的队列的首地址存储到索引表中.

**PCB的产生**:是管理进程的,每产生一个进程就会有一个对应的PCB与之对应产生.

内容:

1. 标识符:
   1. (**PID**)一般是一个数字, 一个进程对应一个PID,唯一的标识符.
   2. 外部标识符:用户访问该进程时使用,用途:计算进程,打印进程,发送进程,接收进程.
   3. 内部标识符:描述进程间的关系,设置用户标识符(识别用户)



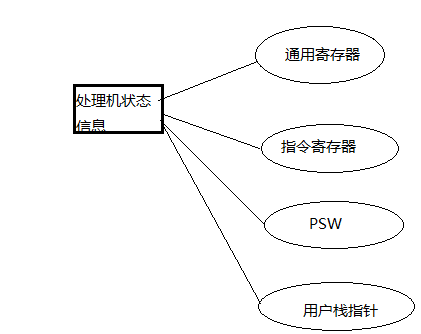
1. 处理机状态信息(方便识别进程状态,将状态相同的进程同一个进程队列中)
   1. 通用寄存器:用户寄存器,用户可访问的寄存器,在其中存储信息

TASK RUNNING( 可运行状态),TASK INTERRUPTIBLE( 可中断的等待状态),TASK UNINTERRUPTIBLE(不可中断的等待状态),TASK ZOMBIE( 僵死状态),TASK STOPPED(暂停状态)

2.2 指令寄存器:下一条指令地址

2.3 程序状态字(PSW):存储进程中的状态信息.

2.4 用户栈指针:存储栈的栈顶指针



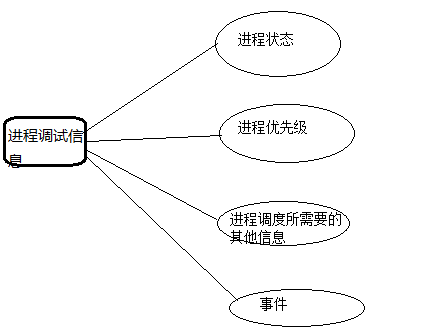
3. 进程调度信息(与进程调度和交换的信息)

3.1 进程状态:进程当前状态

3.2 进程优先级:使用进程调度算法时使用(优先级调度算法)

3.3 进程调度所需要的其他信息

3.4 事件:在进程由进程执行时到进程阻塞时所发生的事情.



4. 进程上下文(与处理器环境相关的信息):是进程执行活动全过程的静态描述

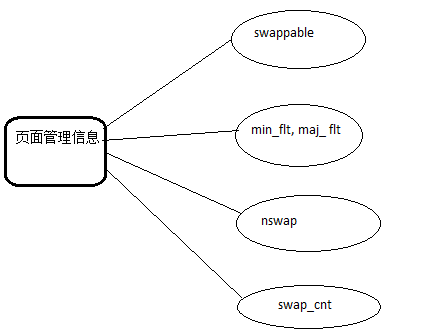
5. 页面管理信息:

5.1该进程的swappable:进程所占用的页面上方可以交换

5.2 min\_flt, maj\_ flt:minor和major的缺页次数

5.3 nswap:该进程累计交换的页面次数

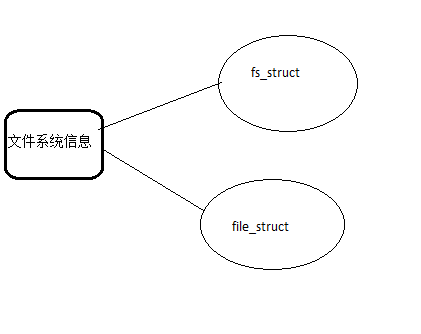
5.4 swap\_cnt:下一次可以交换的页面数



6. 文件系统信息(对进程使用文件的情况进行记录)

6.1 fs\_struct:描述了两个VFS索引节点(可执行映像对应的跟目录和当前目录)

6.2 file\_struc:记录进程打开的文件描述符信息



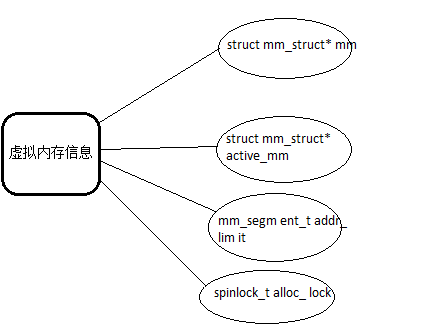
7. 虚拟内存信息

7.1 struct mm\_struct\* mm:描述进程的地址空间

7.2 struct mm\_struct\* active\_mm:内核线程所借用的内存空间

7.3 mm\_segm ent\_t addr\_ lim it:线程空间地址

7.4 spinlock\_t alloc\_ lock:用于申请空间时用的自旋锁(临界区保护)



系统总是通过PCB进行进程的控制,这样很容易知道PCB是进程存在的唯一标识.