1. 我承诺
2. 单核机器中可以使用自旋锁，但是会浪费CPU性能
3. 多核机器不能使用关中断方式形成临界区，因为关中断指令只能关闭一个核的中断
4. 孤儿进程：即自己尚在运行而父亲进程业已退出的进程
5. 僵尸进程：即已经退出但残存资源尚未被父进程回收的进程
6. 父进程在结束自己的执行后不调用wait而是直接exit，则其子进程成为孤儿进程
7. 子进程退出，但父进程未处于调用wait等待子进程退出的状态或父进程没有取得CPU执行，因而子进程残留的资源尚未回收，此时子进程成为僵尸状态
8. 【单队列】：好处在于所有处理机的负载均衡。坏处在于①由于常常需要从一个队列中取任务，故用于互斥同步的开销比较大②缓存亲和力弱，cache难以起作用（每个处理器有自己的cache，故原先由此处理器执行的进程最好还是由其继续执行）

【多队列】：好处在于互斥同步开销小及缓存亲和力强。坏处在于负载不均衡

1. 两级队列，每个处理机有一个自己的队列，并有一个全局队列，当处理机的私有队列中任务执行完毕时，从全局队列中取部分任务加入私有队列。
2. ①内核态产生缺页异常，CPU在执行一条指令后检查中断控制器的中断请求

②收到中断请求，无需切换内核栈故不需要压入ss和esp；将程序使用的eflags, cs, eip压栈。缺页地址保存在CR2寄存器中，利用中断服务例程的段描述符，将其第一条指令的地址加载到cs和eip寄存器中，转到缺页异常处理例程继续执行

③在执行例程中某条指令时再次发生缺页异常，此时无需切换内核栈，只需要把eflags, cs, eip压栈。缺页地址保存在CR2寄存器中，利用中断服务例程的段描述符，将其第一条指令的地址加载到cs和eip中，转到缺页异常处理例程继续执行。

1. 无法计算！60000ms中会发生6000次时钟中断，且固定为10ms一次，因此有另外100次中断发生，但发生的时间节点并不确定，因此无法计算每一个进程在运行时被中断占用了多少时间。
2. Belady指对于给定的访问序列，在采用某种页面置换算法的时候，如果增加可用页面数量，反而会导致缺页次数升高的现象。FIFO和clock算法会产生delady现象。
3. 如下，总缺页次数为6

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 访问  序列 | 0 | 1 | 4 | 2 | 3 | 4 | 1 | 0 | 3 | 2 |
| 物理  页帧  状态 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 |
|  | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
|  |  | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
|  |  |  | 2 | 2 | 2 | 2 | 2 | 2 | 2 |

1. 使用LRU，总缺页次数为7

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 访问  序列 | 0 | 1 | 4 | 2 | 3 | 4 | 1 | 0 | 3 | 2 |
| 物理  页帧  状态 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 |
|  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  |  | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 2 |
|  |  |  | 2 | 2 | 2 | 2 | 0 | 0 | 0 |

1. 并不会一定好。首先，LRU的效果不一定优于FIFO，这取决于访存模式，上面的访问序列就是一个例子；其次，LRU算法的复杂度更高一些，和FIFO相比需要对页面队列/栈有更多的维护操作，因此可能在算法执行时消耗更大。
2. ①bool ptr = false;

②while(TestAndSet(&spin->ptr));

③spin->ptr = false;

1. ①int ptr = 0;

②while(!CompareAndSwap(&spin->ptr, 0, 1));

③spin->ptr = 0;

1. 可以保证公平性，每个进程获取的myturn值是按照顺序依次递增的，而tlock->turn也是递增的，故能保证每个进程按照顺序进入临界区。而自旋锁存在饥饿现象，不能保证访问的公平性。
2. 代码如下：

当A（B）入库时先保证商品数量允许A（B）入库，由enterA（enterB）信号量保证

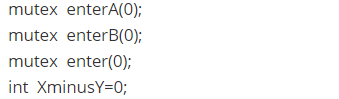
允许入库则申请入库，用enter保证同时只有一件商品在入库操作中

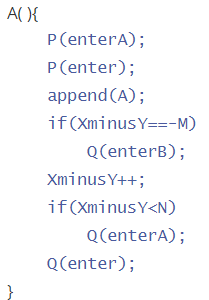
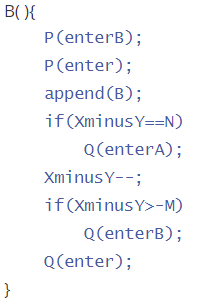
如果此时B（A）商品数量恰过多（少），则允许B（A）入库

增加商品A（B）的数量

如果增加数量后A（B）产品数量尚不过多，则释放enterA（enterB）允许继续入库

释放enter允许其他商品申请入库



1. ①可使用直接映射方式，每个物理页恰对应一个虚拟页。优点是虚拟地址与物理地址相等，容易相互映射；缺点是无法实现页面共享。

②页寄存器，优点是页表大小相对于物理内存来说较小，且页表大小与逻辑地址空间大小无关。缺点是需要根据帧号找页号，且需要对页号进行搜索。

1. 0x1edc8000
2. 0xff3fcff0
3. 0xff22bd2c
4. 0x6f140000
5. ①如果Db完成B[v2]完成，I[v2]未完成，则文件认为自己只有数据块4，相当于数据块5被未知文件占有，因此无法被释放，之后也无法被其他文件使用

②如果Db完成I[v2]完成，B[v2]未完成，则文件认为自己有数据块4和5，但数据块5在bitmap中仍处于未使用状态，有可能被分配给其他文件并被修改