清华大学本科生考试试题专用纸

考试课程：**操作系统（A 卷）**　　时间：2016 年 04 月 06 日下午 3:20~4:55

系别：_____ 班级：_____ 学号：_____ 姓名：

答卷注意事项：
1. 答题前，请先在试题纸和答卷本上写明 A 卷或 B 卷、系别、班级、学号和姓名。
2. 在答卷本上答题时，要写明题号，不必抄题。
3. 答题时，要书写清楚和整洁。
4. 请注意回答所有试题。本试卷有 17 个题目，共 3 页。
5. 考试完毕，必须将试题纸和答卷本一起交回。

一、判断题（20 分）

1. [ ] 如果父进程没有执行 wait()或已退出，那子进程执行 exit()后，将一直是“僵尸进程(zombie)”。
2. [ ] 操作系统中的进程控制块与进程是一一对应关系。
3. [ ] 线程间切换一定比进程间切换快。
4. [ ] 考虑写操作情况的改进时钟页面替换算法不会存在 Belady 现象。
5. [ ] 在 OS 内核中也可以执行系统调用（syscall）来获得 OS 内核的服务。
6. [ ] 一个进程有独立的用户栈和内核栈。
7. [ ] 在 x86 保护模式中，中断/异常/系统调用的服务例程起始地址信息都位于中断描述符表中。
8. [ ] CPU 一直通过直接访问位于内存中页表完成虚实地址转换。
9. [ ] 如果用户态进程一直执行死循环，则操作系统将一直无法执行，从而造成系统“假死”。
10. [ ] 通过页表机制可实现进程间的内存空间隔离和共享。

二、填空题（26 分）

11. 在使能段/页式的 x86 保护模式下，机器指令中出现的内存地址，都是(---11.1---)地址，需要转换成(---11.2---)地址，再经过 MMU(CPU 中的内存管理单元)转换成(---11.3---)地址才能够被访问到。在 Linux for x86 (32 bit)下，我们写个最简单的 hello world 程序，用 gcc 编译，再反编译后会看到以下指令：

```
mov     0x80495b0, %eax
```

这里的内存地址 0x80495b0 就是一个(---11.4---)地址，必须加上隐含的 DS 数据段的基地址，才能构成(---11.5---)地址。也就是说 0x80495b0 是当前进程的 DS 数据段内的偏移。

12. 局部性原理: CPU 访问存储器时，无论是存取指令还是存取数据，所访问的存储单元都趋于聚集在一个较小的连续区域中。有多种不同类型的局部性: (---12.1---)局部性是指一个信息项正在被访问，那么在近期它很可能还会被再次访问。(---12.2---)局部性是指在最近的将来将用到的信息很可能与现在正在使用的信息在空间地址上是临近的。

13. 三状态进程模型中，(---13.1---)状态是指进程正在处理机上运行；(---13.2---)状态是指进程获得了除处理机之外的所有资源，得到处理机即可运行；(---13.3---)状态是指进程正在等待某一事件的

出现而暂停运行的状态。在上述三种状态中，所有处于(---13. 4---)状态的进程都由操作系统内核在相应的进程队列中维护，而处于(---13. 5---)状态的进程不会构成队列。

在实际的进程管理中，进程的状态会更多一些。如果父进程创建子进程并继续先执行 wait(子进程 pid)系统调用后，则父进程将处于(---13. 6---)状态；然后子进程执行，此时子进程处于(---13. 7---)状态；子进程再执行 exit 系统调用后，则子进程将处于(---13. 8---)状态；最后父进程从 wait 系统调用返回到用户态继续执行，则此时子进程已经(---13. 9---)。

14. 在 x86 保护模式下，(---14. 1---)特权级是指当前活动进程的代码段的特权级，并且它定义了当前进程所执行程序的特权级别。(---14. 2---)特权用于描述对应段所属的特权等级，也就是段本身真正的特权级。(---14. 3---)特权级是指进程对段访问的请求权限。

(---14. 4---)特权级的值保存在 CS 段寄存器的最低两位；(---14. 5---)特权级的值存储在段描述符中的权限位；(---14. 5---)特权级的值保存在 DS/FS 等段寄存器的最低两位。

对数据段访问时的特权级控制遵循一个准则：只有相同或更高特权级的代码才能访问相应的数据段。即要求访问数据段的进程执行代码的(---14. 7---)特权级的值 ≤ 待访问的数据段的(---14. 8---)特权级的值，同时进程访问数据所需 的 DS/FS 等段寄存器的(---14. 9---)特权级的值 ≤ 待访问的数据段等的(---14. 10---)特权级的值。

三、问答题（54 分）

15.（18 分）

1）请简要描述物理内存分配算法“伙伴系统（buddy system）”的工作原理，即它是如何维护物理内存的分配状态，以及物理内存分配和物理内存释放操作过程。

2）假定物理内存的初始状态是有一个 1MB 的空闲物理内存块。采用伙伴系统来进行物理内存分配和释放，请给出下面分配和释放序列过程中的物理内存分配状态。

可以用线段，括号等图形方式表示，比如：

----------1M---------- 或 [1M] //表示 1MB 的空闲空间

-----A=512K----- |--------512K------ 或 [A=512K][512K] 表示进程 A 占用了低端 512KB，还剩 512KB

....

a. 进程 A 请求 200KB 物理内存空间；

b. 进程 B 请求 100KB 物理内存空间；

c. 进程 C 请求 50KB 物理内存空间；

d. 进程 D 请求 140KB 物理内存空间；

e. 进程 B 释放已占用的物理内存空间；

f. 进程 A 释放已占用的物理内存空间；

g. 进程 E 请求 110KB 物理内存空间；

h. 进程 C 释放已占用的物理内存空间；

i. 进程 E 释放已占用的物理内存空间；

j. 进程 D 释放已占用的物理内存空间；

16.（18 分）

1）页面替换算法按替换范围分为全局页面替换算法和局部页面替换算法两类，缺页率页面替换算法属于那类？描述缺页率页面替换算法的工作原理。

2）系统物理内存有 6 个物理页帧，进程访问的逻辑（虚拟）地址空间占序号为 0 至 5 的页面。初始状态是所有页面都不在物理内存中，使用缺页率算法时的算法参数“缺页时间间隔”为 2。请下面存储访问序列“2 0 3 2 3 2 1 2 3 5 2”的缺页次数是多少？每次访问后系统中在物理内存中的虚内页号是哪些？请给出计算过程。

17.（18 分）

1）什么是页式存储？页式存储带来的好处有哪些？

2）以图示方式（方框/线条/箭头等+加上必要的属性和转化过程表述的注释）给出页面大小为 4KB 时 X86-32 页式存储系统（不考虑段机制）上的页表结构和虚拟地址到物理地址的转换过程。

3)有一台只有页机制的简化 80386 的 32bit 计算机，有地址范围位 0~256MB 的物理内存空间（physical memory），可表示大小为 256MB，范围为 0xC0000000~0xD0000000 的虚拟地址空间（virtual address space）,页大小（page size）为 4KB，采用二级页表，一个页目录项（page directory entry ，PDE）大小为 4B,一个页表项（page-table entries PTEs）大小为 4B，1 个页目录表大小为 4KB，1 个页表大小为 4KB。

PTE 格式（32 bit） ： 高地址位.....低地址位

```
PFN19 ... PFN0|NOUSE9 ... NOUSE0|WRITABLE|VALID
```

PDE 格式（32 bit） ： 高地址位.....低地址位

```
PT19 ... PT0|NOUSE9 ... NOUSE0|WRITABLE|VALID
```

其中：

NOUSE9 ... NOUSE0 为保留位，要求固定为 0

WRITABLE：1 表示可读写，0 表示只读

VLAID：1 表示有效，0 表示无效

假设 ucore OS 为可读写的两个虚拟地址（即如下所示的两个 va）设置好了虚拟地址<-->物理地址映射（及如下所示的两个 va...,pa...）的二级页表，设置了页目录基址寄存器（page directory base register，PDBR）保存了页目录表的物理地址（按页对齐），其值为 0。已经建立好了从物理地址 0x1000~0x41000 的第二级页表，且页目录表的 index 为 0x300~0x363 的页目录项的(PT19 ... PT0)的值=(index-0x300+3)。 请针对如下所示的<虚拟地址, 物理地址>对，给出对应的页目录项的 index 值,页目录项内容的值，页表项的 index 值，页表项内容的值，即(pde_idx, pde_ctx, pte_idx, pte_cxt)。

```
va 0xc7384bac, pa 0x07141bac
va 0xcaeded27, pa 0x07919d27
```