

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：2018 年 04 月 13 日下午 13:30~15:05

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。本试卷有三类题目，共四页。
 5. 考试完毕，必须将试题纸和答卷本一起交回。

一、对错题 用 V 表示对，用 X 表示错（10 分）

1. ☐ 进程执行系统调用，从用户态切换到内核态执行时，将切换页表和栈。
2. ☐ 进程切换的具体执行过程发生在内核态。
3. ☐ OS 不能让运行在 OS 内核空间的内核线程与用户进程之间进行进程切换。
4. ☐ OS 在建立页表并使能页机制时，需要特权指令才能最终完成。
5. ☐ 如果用户态进程一直执行死循环将导致 OS 内核一直无法控制 CPU。
6. ☐ 二次机会（时钟）页面替换算法有 Belady 异常现象。
7. ☐ OS 内核会直接杀死产生内存访问异常的用户进程。
8. ☐ 由于栈的原因，在 OS 内核中不能执行系统调用（syscall）来获得 OS 内核的服务。
9. ☐ 对于子进程而言，fork() 执行不成功后的返回值 < 0。
10. ☐ 如果不考虑执行性能，ucore os on x86-32 可实现 LRU 页替换算法。

二、填空题（24 分）

小强同学认真上课听讲，参与讨论，并完成了从 lab0~lab3 的所有实验，在学习过程中，了解和学到了很多知识。下面是他的学习心得，请补充完整。

1.1 小强发现完成实验需要在 Linux 下操作很多命令行工具，于是他认真学习了 lab0 中的知识，了解到 Linux 中在命令行模式下可以通过执行命令(__1.1__)来显示当前目录的路径，如果我们编写的程序有语法错误，编译器(__1.2__)会报错，根据错误信息，我们可以修改我们的程序，可以通过硬件模拟器工具(__1.3__)来执行我们的 ucore 操作系统。

1.2 在完成 lab1 的过程中，通过分析硬件模拟器工具对 CPU 状态的输出信息，可了解到基于 80386 的计算机在加电后执行 BIOS 代码时处于(__2.1__)模式。而 os lab1 中的 bootloader 通过建立(__2.2__)表可让计算机进入(__2.3__)模式，从而可以充分利用 80386 CPU 提供的保护能力和 32 位内存寻址空间。os lab1 中的 ucore os 为了能够对异常/中断等进行有效管理，需要建立(__2.4__)表，才能使能中断，让 ucore os 进行进一步的中断处理。在学习 80386 特权级时，对 CPL、RPL 和 DPL 需要满足如下两个公式确保系统安全：访问(__2.5__)时， $CPL \leq DPL[\text{门}] \& CPL \geq DPL[\text{段}]$ ；访问(__2.6__)时， $MAX(CPL, RPL) \leq DPL$ 。

1.3 在完成 lab2 的过程中，需要了解 x86-32 的内存大小与布局，页机制，页表结构等。硬件模拟器

提供了 128MB 的内存，并设定页目录表的起始地址存放(__3.1__)寄存器中，页目录表和页表的地址按(__3.2__)字节对齐。在一个页目录表占用(__3.3__)个 Byte，一个页表占用(__3.4__)个 Byte。ucore 通过 x86-32 CPU 中的(__3.5__)寄存器可以获得发生页面访问错误时的线性地址。

1.4 在完成 lab3 的过程中，ucore 操作系统在页机制基础上，并利用异常机制建立了虚存管理策略与机制。如果一个页（4KB/页）被置换到了硬盘某 8 个连续扇区（0.5KB/扇区），该页对应的页表项（PTE）的最低位--present(存在位)应该为(__4.1__)，表示虚实地址映射关系不存在，而原来用来表示页帧号的高(__4.2__)位，恰好可以用来表示此页在硬盘上的起始扇区的位置（其从第几个扇区开始）。

1.5 在学习进程的概念中，了解到在支持多进程的操作系统（包括 ucore）中，每个进程有两个堆栈，分别是(__5.1__)栈和(__5.2__)栈。操作系统通过建立(__5.3__)这个核心数据结构来支持对进程的管理。对于进程的三状态模型，是指进程在执行过程中会具有(__5.4__)，(__5.5__)，(__5.6__)三种状态。在操作系统具有进程地址空间为单位的 swap in/out 虚存管理机制，可建立进程的五状态模型，将增加(__5.7__)，(__5.8__)。

三、问答题

3.1 （8 分）在 Linux 环境下，下列程序调用 magic 函数的次数是多少？如果一个程序死循环调用 fork()系统调用，会出现什么情况？请说明原因。

```
#include <stdio.h>
#include <unistd.h>
main()
{
    int i;
    for (i = 0; i < 10; i++)
        fork();
    magic();
}
```

3.2 （12 分）用户线程是指由一组用户线程管理库函数来完成线程的管理功能，包括线程的创建、终止、同步和调度等。假设处于仅通过用户线程管理库管理用户线程的操作系统环境，请回答下列问题：

1. 操作系统内核是否需要知道用户线程的存在？请说明理由。
2. 用户线程管理库实现的线程切换是否需要进入内核态，通过操作系统内核来完成？请说明理由。
3. 用户态线程管理库是否可以随时打断用户态线程，完成线程调度与切换？请阐述理由或方法。

3.3 （23 分）在一个只有一级页表的请求页式存储管理系统中，假定页表内容如下表：

页号	页框（Page Frame）号	有效位（存在位）
0	123H	1
1	N/A	0
2	254H	1

页面大小为 4KB，一次内存的访问时间是 100ns，一次快表（TLB）的访问时间是 10ns，处理一

次缺页的平均时间为 10^7 ns（已经包含更新 TLB 和页表的时间），进程的驻留集大小固定为 2，采用最近最少使用置换算法（LRU）和局部淘汰策略。假设①TLB 初始为空；②地址转换时先访问 TLB，若 TLB 没有命中，再访问页表（忽略访问页表之后的 TLB 更新时间）；③有效位为 0 表示页面不在内存，产生缺页中断，缺页中断处理后，返回到产生缺页中断的指令处重新执行。设有虚地址访问序列 2362H，1565H，25A5H，请问：

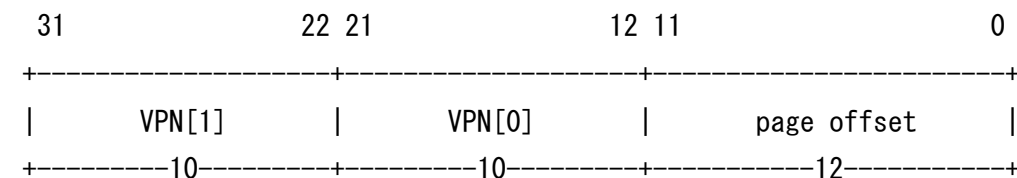
- (1) 依次访问上述三个虚地址，各需要多少时间？给出计算过程？
- (2) 基于上述访问序列，虚地址 1565H 的物理地址是多少？请说明理由。

3.4（23 分） 2017 年图灵奖得主 John L. Hennessy 和 David A. Patterson 提出了 RISC-V 架构的 32 位小端序 CPU 设计，它有 34 位地址总线，使用 32 位页式存储管理。该计算机的页面大小为 4KiB，一个页表大小为 4KiB，其中每一个页表项 (Page Table Entry, PTE) 大小为 4B，虚拟地址、物理地址和 PTE 的结构如下图 1 所示。请回答下列问题：

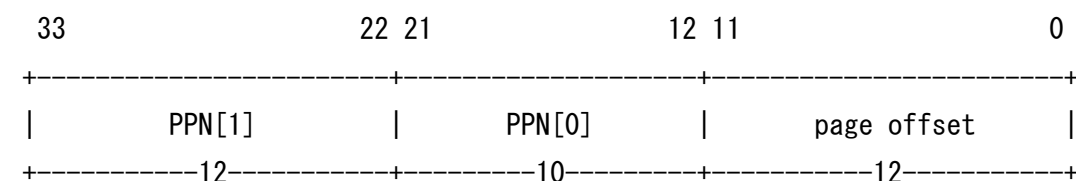
1. 32-bit 的 RISC-V 架构 CPU 使用 34 位物理地址而不是 32 位物理地址，这样做的好处是什么？
2. 设页目录基址为 0x90000000，部分物理内存的内容如下图 2 所示，试给出虚拟地址 0x3A69A4D2 和 0x3A8EB00C 所对应的物理地址和它们所在页的类型。请写出计算过程。

图 1

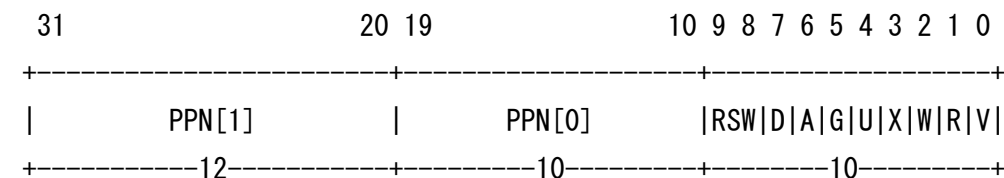
32-bit virtual address:



34-bit physical address:



32-bit page table entry:



如上图所示，一个虚拟地址由虚拟页号 (Virtual Page Number, VPN) 和页内偏移组成，物理地址由物理页号 (Physical Page Number, PPN) 和页内偏移组成，PTE 由 PPN 和一些控制位组成，其中 R/W/X 三个域分别表示对应页的读/写/执行权限，它们的不同组合可以表示不同的属性，如下表所示。

X	W	R	Meaning
0	0	0	This PTE points to next level of page table.
0	0	1	Read-only page.
0	1	0	Reserved for future use.
0	1	1	Read-write page.
1	0	0	Execute-only page.
1	0	1	Read-execute page.
1	1	0	Reserved for future use.
1	1	1	Read-write-execute page.

图 2

0x010000390: 7E B2 B3 12 A0 83 C1 A1 76 F2 0A 77 7F 3A 6A F1
0x0100003A0: 28 AB 4B CC 08 CC CC 0B 00 19 98 07 1A 00 4C 10
0x0100003B0: B9 78 D2 3A C5 C9 A8 ED 16 71 2B F7 96 1C C6 C6
0x0100003C0: BA 4A E1 EC F3 12 D2 0A 58 63 E9 45 2C 5D 26 60
0x0100003D0: 05 4F 45 5E CF E6 D4 FD 3F 90 E2 5E 92 55 C6 87

0x090000380: C7 3E 8C 60 C5 5A 4A 19 69 9E 54 5D BE 9B E8 C5
0x090000390: 85 AA 5A 5E E6 A7 C0 6D 2A 26 3A 72 24 F5 E4 77
0x0900003A0: F7 44 D5 98 01 00 00 28 0F 00 B0 3E 25 15 80 44
0x0900003B0: EB 64 71 8E 1E A4 90 BE 1C E4 E4 4C 62 FA 3A 62
0x0900003C0: 9A 0E 53 3D 2A 52 DD 4F 14 44 12 CA 7C 8A AD 3E

0x0A0000A20: FA 9A EC A1 18 8E B8 EF EB 35 CC EA 87 77 A9 F7
0x0A0000A30: 57 40 02 69 EB C9 5B C6 C0 ED EB F5 1F 44 65 DE
0x0A0000A40: 41 9D 17 19 11 5B 27 D4 7E DE 20 78 33 D1 F6 41
0x0A0000A50: B6 BD E4 E9 15 00 4F 28 48 B4 A6 2E A9 3B D5 BF
0x0A0000A60: 55 EA F7 FC 72 02 17 EA 09 6C AB 37 77 4E 73 BC

0x3C02C0380: 7D D7 2A E6 14 DC 07 F0 6E C5 2E 43 CA E9 20 50
0x3C02C0390: 98 71 3D 43 1B 53 34 F7 FB 8E 64 A1 28 FC 56 36
0x3C02C03A0: 7E 50 9D E9 08 CC CC 0B 00 19 98 07 96 B7 A7 B3
0x3C02C03B0: 2E 79 53 E6 AA 49 56 12 0E B0 50 87 E7 EC 9F B3
0x3C02C03C0: 65 1D 90 64 06 7B 93 C6 96 76 46 4F A9 8F 39 33