ICS Seminar Week11 Prep

康子熙 赵廷昊 余文凯 许珈铭

2023.12.3

Rules

```
remainder <- ordinal number in WeChat Group % 4
for all questions do
    if question number % 4 == remainder then
        you should work on it
    end
end
```

translate

11、假设有一台 64 位的计算机的物理页块大小是 8KB，采用三级页表进行虚拟地址寻址，它的虚拟地址的 VPO（Virtual Page Offset，虚拟页偏移）有 13 位，问它的虚拟地址的 VPN（Virtual Page Number，虚拟页号码）有多少位？

答：（　　）

A. 20　　B. 27　　C. 30　　D. 33

C

14.对于虚拟存储系统，一次访存过程中，下列命中组合不可能发生的是(　　)

A. TLB 未命中，Cache 未命中，Page 未命中

B. TLB 未命中，Cache 命中，Page 命中

C. TLB 命中，Cache 未命中，Page 命中

D. TLB 命中，Cache 命中，Page 未命中

D

16.为使虚拟内存系统有效发挥预期作用，所运行的程序应该具有的特点是
(　　)

A．该程序不应该含有过多的 I/O 操作

B．该程序的大小不应超过实际的内存容量

C．该程序应具有较好的局部性

D．该程序的指令相关不应过多

C

14．虚拟内存管理方式可行性的基础是：

A． 程序执行的离散性

B． 程序执行的顺序性

C． 程序执行的局部性

D． 程序执行的并发性

C

15. Intel 的 IA32 体系结构采用二级页表，称第一级页表为页目录（Page Directory），第二级页表为页表（Page Table）。页面的大小为 4KB，页表项 4 字节。以下给出了页目录与若干页表中的部分内容，例如，页目录中的第 1 个项索引到的是页表 3，页表 1 中的第 3 个项索引到的是物理地址中的第 5 个页。则十六进制逻辑地址 8052CB 经过地址转换后形成的物理地址应为十进制的（）

页目录		页表 1		页表 2		页表 3	
VPN	页表号	VPN	页号	VPN	页号	VPN	页号
1	3	3	5	2	1	2	9
2	1	4	2	4	4	3	8
3	2	5	7	8	6	5	3

A. 21195
B. 29387
C. 21126
D. 47195

B

16. 已知某系统页面长 8KB，页表项 4 字节，采用多层分页策略映射 64 位虚拟地址空间。若限定最高层页表占 1 页，则它可以采用多少层的分页策略？

A. 3 层

B. 4 层

C. 5 层

D. 6 层

C

12. 在 Core i7 中，以下哪个页表项属于 4 级页表项，不属于 1 级页表项：

A. G 位（Global Bit）

B. D 位（Dirty Bit）

C. XD 位（Disable or enable instruction fetch）

D. U/S 位（User or supervisor mode access permission）

B

13. 在 Core i7 中，关于虚拟地址和物理地址的说法，不正确的是：

A. VPO = CI + CO

B. PPN = TLBT + TLBI

C. VPN1 = VPN2 = VPN3 = VPN4

D. TLBT + TLBI = VPN

B

8. 假定整型变量 A 的虚拟地址空间为 0x12345cf0，另一整形变量 B 的虚拟地址 0x12345d98，假定一个 page 的长度为 0x1000 byte，A 的物理地址数值和 B 的物理地址数值关系应该为：

A. A 的物理地址数值始终大于 B 的物理地址数值

B. A 的物理地址数值始终小于 B 的物理地址数值

C. A 的物理地址数值和 B 的物理地址数值大小取决于动态内存分配策略，

D. 无法判定两个物理地址值的大小

B

9. 虚拟内存中两层页表和单层页表相比，最主要的优势在于：

A. 更快的地址翻译速度

B. 能够提供页面更加精细的保护措施

C. 能够充分利用代码的空间局部性

D. 能够充分利用稀疏的内存使用模式

D

10. 根据本课程介绍的 Intel x86-64 存储系统，填写表格中某一个进程从用户态切换至内核态时，和进程切换时对 TLB 和 cache 是否必须刷新。

A. ①不必刷新 ②不必刷新 ③刷新 ④不必刷新

B. ①不必刷新 ②不必刷新 ③不必刷新 ④不必刷新

C. ①刷新 ②不必刷新 ③刷新 ④刷新

D. ①刷新 ②不必刷新 ③不必刷新 ④刷新

A

11. 已知某系统页面长 2KB，页表项 8 字节，采用多层分页策略映射 48 位虚拟地址空间。若限定最高层页表占 1 页，则它可以采用多少层的分页策略？

A．3 层

B．4 层

C．5 层

D．6 层

C

13. 在一个具有 TLB 和高速缓存的系统中，假设地址翻译使用四级页表来进行，且不发生缺页异常，那么在 CPU 访问某个虚拟内存地址的过程中，至少会访问（ ）次物理内存，至多会访问（ ）次物理内存。

A. 0，4

B. 0，5

C. 1，4

D. 1，5

B

14. 假设在 64 位系统下页大小被设置为 16KB，那么采用类似 Core i7 的地址翻译过程，四级页表最多可以索引多少位地址空间？

A. 64

B. 58

C. 54

D. 52

B

15. 为了支持 16G 的虚拟地址空间，采用 3 级页表，页大小为 1KB，页表项大小依旧为 4 字节。现在映射总大小为 1MB 的虚拟地址，其分布未知但分布的最小单位限定为 1 字节，请问实现上述映射的页表结构占用的内存至少至多分别为多大?

A. 6KB，4113KB

B. 6KB，65793KB

C. 6KB，1052688KB

D. 3KB，4113KB

B

16. 某 64 位系统，物理内存地址长度为 52 位，虚拟内存地址长度为 43 位，已知页的大小为 8KB，采用多级页表进行地址翻译，每级页表都占一页，则需要几级页表：

A. 2 级　　B. 3 级　　C. 4 级　　D. 5 级

B

17. 在页表条目中，以下哪个条目是由 MMU 在读和写时进行设置，而由软件负责清除：

A. P 位，子页或子页表是否在物理内存中

B. G 位，是否为全局页（在任务切换时，不从 TLB 中驱逐出去）

C. CD 位，能/不能缓存子页或子页表

D. D 位，修改位，是否对子页进行了修改操作

D

1. 下列关于虚存和缓存的说法中，**正确**的是：

A. TLB 是基于物理地址索引的高速缓存

B. 多数系统中，SRAM 高速缓存基于虚拟地址索引

C. 在进行**线程**切换后，TLB 条目绝大部分会失效

D. 多数系统中，在进行进程切换后，SRAM 高速缓存中的内容不会失效

D

1. 虚拟内存为内存的使用和管理提供了简化，这样的简化**没有**体现在

A. 编译器将 C 文件编译为目标文件的过程

B. 链接器生成完全链接的可执行文件的过程

C. 加载器向内存中加载可执行文件和共享对象文件的过程

D. 不同进程共享相同物理页面的过程

A

2. 下列选项中**错误**的是

A．在使用虚拟地址空间的的系统中，程序引用的页面总数**必须不超过**物理内存总大小。

B．**主存中**的每个**有效**字节都有至少一个选自虚拟地址空间的**虚拟地址**和一个选自物理地址空间的**物理地址**。

C．当在程序中正常调用 malloc 函数时，操作系统会分配出相应大小（例如 k 个）的**连续虚拟页面**，并且将它们映射到物理内存中**任意位置**的 k 个**物理页面**。

D．**不同**进程的多个**虚拟**页面可以映射到**同一个**共享**物理**页面上。

A

4. 以下关于虚拟内存的说法**错误**的是

A. 虚拟内存一般不需要来自应用程序开发者的干涉

B. 虚拟地址空间可以比物理内存更小

C. 连续的虚拟内存**总是**映射到连续的物理内存

D. 目标文件中的.bss 段映射到全是二进制零的匿名文件

C

map

12、进程 P1 通过 fork()函数产生一个子进程 P2。假设执行 fork()函数之前，进程 P1 占用了 53 个（用户态的）物理页，则 fork 函数之后，进程 P1 和进程 P2 共占用_____个（用户态的）物理页；假设执行 fork()函数之前进程 P1 中有一个可读写的物理页，则执行 fork()函数之后，进程 P1 对该物理页的页表项权限为_____。上述两个空格对应内容应该是（　　）

A．53，读写　B．53，只读　C．106，读写　D．106，只读

B

13、下列哪个例子是外部碎片？答：（　　）

A．分配块时为了字节对齐而多分配的空间

B．空闲块中互相指向的指针所占据的空间

C．多次释放后形成的不连续空闲块

D．用户分配后却从未释放的堆空间

C

14、用带有 header 和 footer 的隐式空闲链表实现分配器时，如果一个应用请求一个 3 字节的块，下列说法哪一项是错误的？答：（　　）

A．搜索空闲链表时，存储利用率为：best fit > next fit > first fit

B．搜索空闲链表时，吞吐率为：next fit > first fit > best fit

C．在 x86 机器上，malloc(3)实际分配的空闲块大小可能为 8 字节

D．在 x64 机器上，malloc(3)返回的地址可能为 2147549777

D

15. 有程序段如下：

```
int foo( ) {
    char str1[20], *str2;
    str2 = (char*)malloc(20*sizeof(char));
    free(str2);
}
```

下列说法中正确的是(　　)

A. str1 和 str2 指向的内存都是分配在栈空间内的

B. str1 和 str2 指向的内存都是分配在堆空间内的

C. str1 指向的内存是分配在栈空间内的，str2 指向的内存是分配在堆空间内的

D. str1 指向的内存是分配在堆空间内的，str2 指向的内存是分配在栈空间内的

C

17.动态内存管理中，可能会造成空闲链表中，小空闲块，即“碎片”，比较集中的算法是(　　)

A．首次适配算法　　B．下次适配算法

C．最佳适配算法　　D．以上三种算法无明显区别

A

13. 动态管理器分配策略中，最适合“最佳适配算法”的空白区组织方式是：

A. 按大小递减顺序排列

B. 按大小递增顺序排列

C. 按地址由小到大排列

D. 按地址由大到小排列

B

11．在 C 语言中实现 Mark-and-Sweep 算法时，可以基于以下哪个假设：（宿主机为 32 位机器）

A．所有指针指向一个块的起始地址

B．所有指针数据都是 4 字节对齐

C．只需要扫描数据类型为指针的堆中的数据空间

D．只需要扫描所有长度为 4 字节的堆中的数据空间

B

11．关于动态内存分配，下列说法中正确的是：

A．显式分配器可以重新排列请求顺序，从而最大化内存利用率

B．显式分配器可以修改已分配的块，把内容复制到别的位置，从而消除外部碎片

C．通常显式分配器会比隐式分配器更快

D．C 语言中如果某个已分配块不再可达，那它就会被释放并返回给空闲链表

12．在设计分配器时，下列说法中错误的是：

A．搜索空闲链表时，存储利用率为：best fit > next fit > first fit

B．带头部的隐式空闲链表，合并（内存中的）下一个空闲块可在常数时间内完成

C．如果采取立刻合并策略（immediate coalescing），会在某些请求模式中出现反复合并又分割的情况，于是会有较小的吞吐率

D．分配器使用二叉树结构，主要是为了能够更快地找到适配的空闲块

13．下列说法错误的是：

A．虚拟内存存放在磁盘上，这导致不命中时处罚非常大

B．使用 fork 函数时，不会立刻拷贝内存空间

C．进程不会意外访问别的进程的内存空间

D．使用动态内存的主要原因是栈容易受到缓存溢出（buffer overflow）的腐蚀

7. 下列与虚拟内存有关的说法中哪些是不对的？

A. 操作系统为每个进程提供一个独立的页表，用于将其虚拟地址空间映射到物理地址空间。

B. MMU 使用页表进行地址翻译时，虚拟地址的虚拟页面偏移与物理地址的物理页面偏移是相同的。

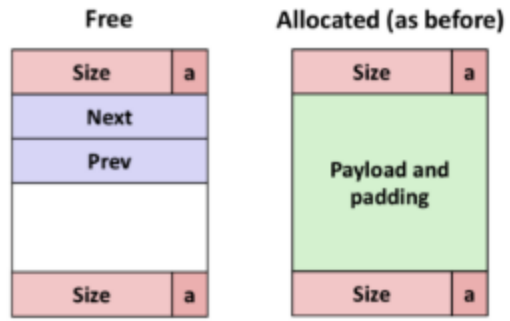
C. 若某个进程的工作集大小超出了物理内存的大小，则可能出现抖动现象。

D. 动态内存分配管理，采用双向链表组织空闲块，使得首次适配的分配与释放均是空闲块数量的线性时间。

D

12. 现在有一个用户程序执行了如下调用序列

```
void *p1 = malloc(16);
void *p2 = malloc(32);
void *p3 = malloc(32);
void *p4 = malloc(48);
free(p2);
void *p5 = malloc(4);
free(p3);
void *p6 = malloc(56);
void *p7 = malloc(10);
```



内存分配器内部使用**显式空闲链表**实现，按照地址从低到高顺序来维护空闲链表节点顺序。分配块格式参照上图。每个分配块 16 字节对齐，头部和脚部的大小都是 4 字节。分配算法采用**首次适配**算法，将适配到的空闲块的第一部分作为分配块，剩余部分变成新的空闲块，并采用**立即合并**策略。假设初始空闲块的大小是 1KB。那么以上调用序列完成后，分配器管理的这 1KB 内存区域中，**内部碎片**的总大小是____，链表里**第一个**空闲块的大小是_____。

A. 58Byte，848Byte

B. 26Byte，32Byte

C. 74Byte，48Byte

D. 74Byte，16Byte

D

18. 关于写时复制（copy-on-write）技术的说法，不正确的是：

A. 写时复制既可以发生在父子进程之间，也可以发生在对等线程之间

B. 写时复制既需要硬件的异常机制，也需要操作系统软件的配合

C. 写时复制既可以用于普通文件，也可以用于匿名文件

D. 写时复制既可以用于共享区域，也可以用于私有区域

A

19. 垃圾收集器的根节点不包括以下哪个节里的数据:

A. text
B. data
C. bss
D. stack

A

2. 阅读下列代码并回答选项。（已知文件“input.txt”中的内容为“12”，头文件没有列出）

```
void *Mmap(void *addr, size_t length, int prot, int flags,
           int fd, off_t offset);

int main(){
    int status;
    int fd = Open("./input.txt", O_RDWR);
    char* bufp = Mmap(NULL, 2, PROT_READ | PROT_WRITE ,
                      MAP_PRIVATE, fd, 0);
    if (Fork()>0){
       while(waitpid(-1,&status,0)>0);
       *(bufp+1) = '1';
       Write(1, bufp, 2); // 1: stdout
       bufp = Mmap(NULL, 2, PROT_READ, MAP_PRIVATE, fd, 0);
       Write(1, bufp, 2);
    }
    else{
        *bufp = '2';
        Write(1, bufp, 2);
    }
```

在 shell 中运行该程序，正常运行时的终端输出应为

A. 221112　B. 222121　C. 222112　D. 221111

A

3. 以下关于动态内存分配的说法中，**错误**的是

A. 可以通过调用 sbrk(0) 获取当前进程中堆的顶部地址

B. 如果向一个已经 free 的指针写入数据，**一定会触**发异常

C. 如果使用 malloc(0x10) 获取一个指针，然后写入 0x200 字节大小的数据，**不一定**会触发异常

D. 使用 mmap 也是动态分配内存的方法之一

B

第五题（12 分）

1. （2 分，每空一分，考察多级页表和单级页表的设计思想的理解）

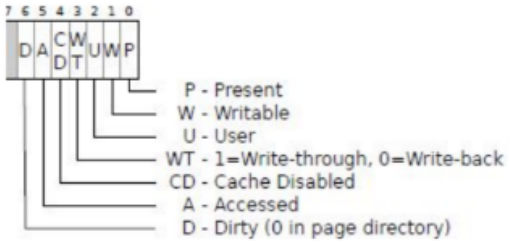
在进行地址翻译的过程中，操作系统需要借助页表(Page Table)的帮助。考虑一个 32 位的系统，页大小是 4KB，页表项(Page Table Entry)大小是 4 字节(Byte)，如果不使用多级页表，常驻内存的页表一共需要_____页。

考虑下图已经显示的物理内存分配情况，在二级页表的情况下，已经显示的区域的页表需要占据_____页。

<table>
<tr><td>VP0</td><td rowspan="6">已分配页</td></tr>
<tr><td>…</td></tr>
<tr><td>VP1023</td></tr>
<tr><td>VP1024</td></tr>
<tr><td>…</td></tr>
<tr><td>VP2047</td></tr>
<tr><td>Gap</td><td rowspan="2">未分配页</td></tr>
<tr><td>1023
unallocated
pages</td></tr>
<tr><td>VP10239</td><td rowspan="4">已分配页</td></tr>
<tr><td>VP10240</td></tr>
<tr><td>…</td></tr>
<tr><td>VP11263</td></tr>
</table>

2. （6 分，每错一空扣一分，扣完为止。考察地址翻译过程）

IA32 体系采用小端法和二级页表。其中两级页表大小相同，页大小均为 4KB，结构也相同。TLB 采用直接映射。TLB 和页表每一项的后 7 位含义如下图所示。为简便起见，假设 TLB 和页表每一项的后 8~12 位都是 0 且不会被改变。注意后 7 位值为“27”则表示可读写。



当系统运行到某一时刻时，TLB 内容如下：

索引	TLB 标记	内容	有效位
0	0x04013	0x3312D027	1
1	0x01000	0x24833020	0
2	0x005AE	0x00055004	1
3	0x00402	0x24AEE020	0
4	0x0AA00	0x0005505C	0
5	0x0000A	0x29DEE000	1
6	0x1AE82	0x00A23027	1
7	0x28DFC	0x00023000	0

一级页表的基地址为 0x0C23B00，物理内存中的部分内容如下：

地址	内容	地址	内容	地址	内容	地址	内容
00023000	E0	00023001	BE	00023002	EF	00023003	BE
00023120	83	00023121	C8	00023122	FD	00023123	12
00023200	23	00023201	FD	00023202	BC	00023203	DE
00023320	33	00023321	29	00023322	E5	00023323	D2
0005545C	97	0005545D	C2	0005545E	7B	0005545F	45
00055464	97	00055465	D2	00055466	7B	00055467	45
0C23B020	27	0C23B021	EB	0C23B022	AE	0C23B023	24
0C23B040	27	0C23B041	40	0C23B042	DE	0C23B043	29
0C23B080	05	0C23B081	5D	0C23B082	05	0C23B083	00
2314D200	23	2314D201	12	2314D202	DC	2314D203	0F
2314D220	A9	2314D221	45	2314D222	13	2314D223	D2
29DE404C	27	29DE404D	42	29DE404E	BA	29DE404F	00
29DE4400	D0	29DE4401	5C	29DE4402	B4	29DE4403	2A

此刻，系统先后试图对两个已经缓存在 cache 中的内存地址进行写操作，请分析**完成写之后**系统的状态（写的地址和上面的内存地址无交集），完成下面的填空。若不需要某次访问或者缺少所需信息，请填“\”。

第一次向地址 0xD7416560 写入内容，TLB 索引为：_____，完成写之后该项 TLB 内容为：_____，

二级页表页表项地址为：_____，物理地址为：_____。

第二次向地址 0x0401369B 写入内容，

TLB 索引为：_____，完成写之后该项 TLB 内容为：_____

二级页表页表项地址为：_____，物理地址为：_____。

3. (2 分，考察对于虚拟内存独立地址空间的理解)

本学期的 fork bomb 作业中，大家曾用 fork 逼近系统的进程数量上限。下面有一个类似的程序，请仔细阅读程序并填空。

```
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
#define N 4
int main() {
        volatile int pid, cnt = 1;
        for (int i = 0; i < N; i++) {
                if ((pid = fork()) > 0) {
                        cnt++;
        }
        while (wait(NULL) > 0);
        return 0;
}
```

整个过程中，变量 cnt 最大值是_____。假设所有的数据都已经存在于内存中，pid 和 cnt 在同一个物理页中。从第一个进程开始执行 for 语句开始，此过程对于 cnt 的操作至少会导致页表中_____次虚拟页对应的物理页被修改。

1024, 5

6, 0x00A23067, \, 0x00A23560

3, 0x00BA4067, 0x29DE404C, 0x00BA469B

5, 15