装订线内

不要答题

北京大学信息科学技术学院考试试卷

考试科目： 计算机系统导论 **姓名：** _____ **学号：** _____

考试时间： 2014 年 1 月 7 日 **任课教师：** _____

题号	一	二	三	四	五	六	七	八	**总分**
分数									
阅卷人									

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机、或有存储、编程、查询功能的电子用品等）不得带入座位，已经带入考场的必须放在监考人员指定的位置。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束时间到，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准交头接耳，不准偷看、夹带、抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有违纪作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学术规范条例》及相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共　　页。

得分

第一题 单项选择题（每小题 1.5 分，共 30 分）

1、对于 IEEE 浮点数，如果减少 1 位指数位，将其用于小数部分，将会有怎样的效果？答：（ ）

A．能表示更多数量的实数值，但实数值取值范围比原来小了。

B．能表示的实数数量没有变化，但数值的精度更高了。

C．能表示的最大实数变小，最小的实数变大，但数值的精度更高。

D．以上说法都不正确。

答案：C

因为 NaN 更多，所以能表示的实数值更少。

指数小了，所以最大实数变小，最小实数变大。

小数部分增加了，所以精度更高。

2、按照教材描述的原则，对于 x86_64 程序，在 callq 指令执行后，函数的第一个参数一般存放在哪里？答：（ ）

A．8(%rsp) B．4(%rsp) C．%rax D．%rdi

答案：D

考察学生对 x86_64 与 IA-32 在函数参数传递上的不同。

3、已知变量 x 的值已经存放在寄存器 eax 中，现在想把 5x+7 的值计算出来并存放到寄存器 ebx 中，如果不允许用乘法和除法指令，则至少需要多少条 IA-32 指令完成该任务？答：（ ）

A．1 条 B．3 条 C．2 条 D．4 条

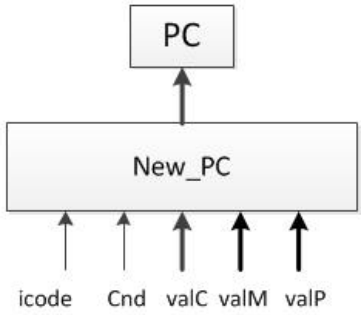
答案：A

考察学生对 leal 指令的用法是否熟悉。

使用一条指令“leal 7(%eax, %eax, 4), %ebx”即可。

4、 在 Y86 的 SEQ 实现中，PC（Program Counter，程序计数器）更新的逻辑结构如下图所示，请根据 HCL 描述为①②③④选择正确的数据来源。

```
Int new_pc = [
    # Call.
    Icode = ICALL : ①;
    # Taken branch.
    Icode = IJXX && Cnd : ②;
    # Completion of RET instruction.
    Icode = IRET : ③;
    # Default.
    1 : ④;
```



其中：Icode 为指令类型，Cnd 为条件是否成立，valC 表示指令中的常数值，valM 表示来自返回栈的数据，valP 表示 PC 自增。

① **A** ：A）valC B）valM C）valP

② **A** ：A）valC B）valM C）valP

③ **B** ：A）valC B）valM C）valP

④ **C** ：A）valC B）valM C）valP

（5~6） 如果直接映射高速缓存（Cache）的大小是 4KB，并且块大小（block）大小为 32 字节。

5、请问它每路（way）有多少行（line）？ 答：（ ）

A. 128 B. 64 C. 32 D. 1

答案：选 A，容量=路容量*路数=cache 行大小*Set 数*路数->set 数=4KB/32B=2^7=128

6、如果数据访问的地址序列为 0->4->16->132->232->4096->160（以字节为单位），请问一共发生多少次替换？答：（ ）

A. 0 B. 1 C. 2 D. 3

答案：选 B，连续访问 set 数相同的数据会发生替换，set 号有地址 5~11 位决定，等于地址 A mod(128*32)，只有 0 与 4096 位于同一 cache 行

7、下列程序运行的结果是什么？答：（ ）

```
/* main.c */
int i=0;
int main()
{
  foo();
  return 0;
}

/* foo.c */
int i=1;
```

```
void foo()
{
  printf("%d", i);
}
```

A．编译错误　　B．链接错误

C．段错误　　D．有时打印输出 1，有时打印输出 0；

答案：B 考点：链接的基本概念，符号解析的原则；

注：由于印刷版中引号的问题（现已修正），本题选 A 也算对

8、在链接时，对于什么样的符号一定不需要进行重定位？答：（　　）

A．不同 C 语言源文件中定义的函数

B．同一 C 语言源文件中定义的全局变量

C．同一函数中定义时不带 static 的变量

D．同一函数中定义时带有 static 的变量

答案：C

说明：考察需要进行重定位的条件。A 在链接前不在同一目标文件中，BD 都位于.data 段中，这些都需要重定位。C 位于栈中或者寄存器中，不需要重定位。

9、关于信号的描述，以下不正确的是哪一个？答：（　　）

A．在任何时刻，一种类型至多只会有一个待处理信号

B．信号既可以发送给一个进程，也可以发送给一个进程组

C．SIGTERM 和 SIGKILL 信号既不能被捕获，也不能被忽略

D．当进程在前台运行时，键入 Ctrl-C，内核就会发送一个 SIGINT 信号给这个前台进程

答案：C

说明：考察信号的基本了解，C 应该为“SIGKILL 和 SIGSTOP 信号既不能被捕获，也不能被忽略”

10、下面关于非局部跳转的描述，正确的是（　　）

A．setjmp 可以和 siglongjmp 使用同一个 jmp_buf 变量

B．setjmp 必须放在 main()函数中调用

C．虽然 longjmp 通常不会出错，但仍然需要对其返回值进行出错判断

D．在同一个函数中既可以出现 setjmp，也可以出现 longjmp

答案：D

说明：考察非局部跳转的基本了解；D 选项：可以采用不同的 jmp_buf

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

答案：选 C。

本题考查对页表组成的理解。

页块大小为 8KB，即 2^13 byte。在 64 位机器上，一个页表条目为 8byte。故共有页表条目 2^10 项，故每一级页表可以表示 10 位地址。因此三级页表存储，共需要 10*3=30 位。

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

答案：选 B。

fork 使用之后，只生成新的 mm_struct 等结构，不会将原来的数据也拷贝一份，而是采用了 copy-on-write 的策略，因而占用的物理内存不会是原来的两倍，同时这两个进程的页表项权限都只有读。

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

答案：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

答案 ：D，malloc 在 64 位机器上返回的地址应该按 16 字节对齐。

15、考虑如下代码，假设 result.txt 的初始内容是“123”。

```
int main(int argc, char** argv)
{
    int fd1 = open("result.txt", O_RDWR);
    char str[] = "abc";
    char c;
    write(fd1, str, 1);
    read(fd1, &c, 1);
    write(fd1, &c, 1);
    return 0;
}
```

在这段代码执行完毕之后，result.txt 的内容是什么？（假设所有的系统调用都会成功）答：（　　）

A. a22　　B. a21　　C. a13　　D. abb

答案：A。考察文件操作过程中指针的移动情况。

16、已知如下代码段

```
write(fd1, str1, strlen(str1));
write(fd2, str2, strlen(str2));
```

可以在原本为空的文件 ICS.txt 中写下字符串 I love ICS!

对于下面这些对于变量 fd1, fd2, str1, str2 的定义：

```
(1)
int fd1 = open("ICS.txt", O_RDWR);
int fd2 = open("ICS.txt", O_RDWR);
char *str1 = "I love ";
char *str2 = "ICS!";
(2)
int fd1 = open("ICS.txt", O_RDWR);
int fd2 = dup(fd1);
char *str1 = "I love ";
char *str2 = "ICS!";
(3)
int fd1 = open("ICS.txt", O_RDWR);
int fd2 = open("ICS.txt", O_RDWR);
char *str1 = "I love ";
char *str2 = "I love ICS!";
(4)
int fd1 = open("ICS.txt", O_RDWR);
int fd2 = dup(fd1);
```

```
char *str1 = "I love ";
char *str2 = "I love ICS!";
```

下面哪一个组合是正确的：（　　）

A. (1)(4)　　B. (2)(3)　　C. (1)(2)(3)(4)　　D. 都不正确

答案：B

说明：考察两种不同文件描述符指向同一文件 v 的不同，一种是指向了同一的 open file table，一种是指向了同一的 v-node table。

17、下列关于计算机网络概念的说法中，哪一项是正确的？答：（　　）

A. 全球最大的计算机网络是互联网 Internet，所以计算机网络协议是 Internet Protocol 即 IP 协议。

B. 计算机之间的网络通信是一个机器上的一个 process（如 client process）与另一个机器上的 process（如 server process）之间的通信。

C. 网络应用程序有默认的端口号，大部分应用的端口号可以修改，而少部分知名应用如 Web 服务程序的端口号 80 是无法修改的。

D. 一个域名只能对应一个 IP 地址；而一个 IP 地址可以对应多个域名。

答案：B

解释：考察第一部分

A. IP 只是计算机网络众多协议中的一种；IP 是具有代表性的一种协议；但还有其他很多协议。

C. 端口号是可以修改的，默认端口号只是为了方便

D. 一个域名可以对应多个 IP；一个 IP 也可以对应多个域名。

18、在 client-server 模型中，一个连接（connection）可以由 IP 地址，端口号的组合来表示。假设一个访问网页服务器的应用，客户端 IP 地址为 128.2.194.242，目标服务器端 IP 地址为 208.216.181.15，用户设置的代理服务器 IP 地址为 155.232.108.39。目标服务器同时提供网页服务（默认端口 80），和邮件服务（默认端口 25）。当客户端向目标服务器发送访问网页的请求时，下面 connection socket pairs 正确的一组是？答：（　　）

	客户端请求	代理请求
A	(128.2.194.242:25, 155.232.108.39:80)	(128.2.194.242:51213, 208.216.181.15:80)
B	(128.2.194.242:51213, 155.232.108.39:80)	(128.2.194.242:12306, 208.216.181.15:80)
C	(128.2.194.242:25,	(155.232.108.39:51213,

	208.216.181.15:80)	208.216.181.15:80)
D	(128.2.194.242:51213, 208.216.181.15:80)	(155.232.108.39:12306, 208.216.181.15:80)

答案：D

解释：考察所有三部分内容

（客户端请求）这一段，客户端发送的请求，源地址是客户自己地址 128.2.194.242，端口号是临时端口号；目标地址仍是服务器地址 208.216.181.15:80 默认端口 80

（代理请求）这一段，代理发送的请求，源地址是代理自己地址 155.232.108.39，端口号是临时端口号；目标地址仍是服务器地址 208.216.181.15:80 默认端口 80。

所以，选项 D 正确。

（选项 A/B 中，目标地址不是代理的地址，而是目标服务器的地址。）

（选项 C 中，端口号 25 属于电子邮件应用默认端口号，不作为客户端的临时端口号。）

19、在 Pthread 线程包使用中，下列代码输出正确的是：（　　）

```
void *th_f(void * arg)
{
   printf("Hello World") ;
   pthread_exit(0) ;
 }

int main(void)
{
   pthread_t tid;
   int st;
   st = pthread_create(&tid,NULL,th_f,NULL);
   if(st<0) {
      printf("Oops, I can not create thread\n");
      exit(-1);
   }
   sleep(1);
   exit(0);
}
```

A. Oops, I can not create thread

B. Hello World

Oops, I can not create thread

C. Hello World

D. 不输出任何信息

答案：C

20、有四个信号量，初值分别为：a=1， b=1， c=1， d=1

线程 1：	线程 2：	线程 3：
① P(a);	⑨ P(c);	⑮ P(d);
② P(b);	⑩ P(b);	⑯ P(a);
③ P(d);	⑪ V(c);	⑰ V(d);
④ P(c);	⑫ V(b);	⑱ P(b);
⑤ V(d);	⑬ P(a);	⑲ V(a);
⑥ V(a);	⑭ V(a);	⑳ V(b);
⑦ V(b);		
⑧ V(c);		

上面的程序执行时，下列哪些执行轨迹不会产生死锁？

A. ①②③⑨⑩⑮④　　B. ⑨①②③⑩

C. ①②⑮⑨③　　D. ⑨①⑮②③

答案：实际上四个选项都有问题，无正确答案，因此批卷时本题直接算作得分

得分

第二题（8 分）

考虑一种新的遵从 IEEE 规范的浮点的格式，包含 3 位指数位，4 位小数位和 1 位符号位。请填写下面的表格。

描述	十进制数(或分数)	二进制表示
Bias	**3**	--------
最小的正数	**1/64**	**0 000 0001**
最小的有限数	**-15.5**	**1 110 1111**
最小规格正数	1/4	**0 001 0000**
---	-11/32	**1 001 0110**
---	7/2	**0 100 1100**
---	63/128	**0 010 0000**
---	13.25	**0 110 1010**

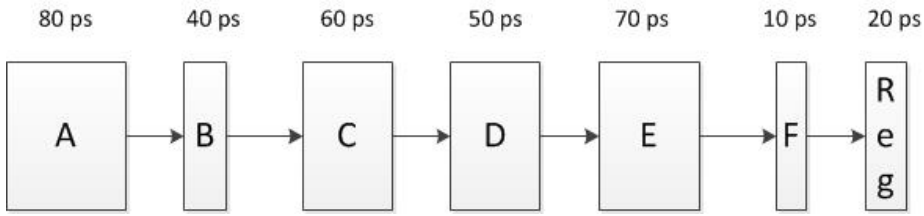
每行 1 分

考察学生对浮点数的掌握程度，包括舍入。

得分

第三题 （12 分）

如图所示，每个模块表示一个单独的组合逻辑单元，每个单元的延迟已在图中标出。通过在两个单元间添加寄存器的方式，可以对该数据通路进行流水化改造。假设每个寄存器的延迟为 20ps。



1）如果改造为一个二级流水线（只插入一个寄存器），为获得最大的吞吐率，该寄存器应在哪里插入？请计算该流水线的吞吐率，并说明计算过程。

插入在 CD 间（1 分）

1000/(80+40+60+20) = 1000/200= 5GIPS（过程 1 分，结果 1 分）

2）如果改造为一个三级流水线（插入两个寄存器），为获得最大的吞吐率，寄存器应在哪里插入？请计算该流水线的吞吐率，并说明计算过程。

插入在 BC 间和 DE 间（1 分）

1000/(80+40+20) = 1000/140 = 7.143 GIPS（过程 1 分，结果 1 分）

3）如果改造为一个四级流水线（插入三个寄存器），为获得最大的吞吐率，寄存器应在哪里插入？请计算该流水线的吞吐率，并说明计算过程。

插入 AB 间、CD 间、DE 间（1 分）

1000/(100+20) = 1000/120 = 8.33 GIPS（过程 1 分，结果 1 分）

4）不改变单元划分，为获得最大性能，该设计至少需要划分成几级？请计算对应的吞吐率，并说明计算过程。

至少划分成 5 级（1 分）

1000/(80+20) = 10 GIPS（过程 1 分，结果 1 分）

得分

第四题（10 分）

考虑如下两个程序（fact1.c和fact2.c）：

```
/* fact1.c */
#define MAXNUM 12
int table[MAXNUM];
int fact(int n);
int main(int argc, char **argv) {
    int n;
    table[0] = 0;
    table[1] = 1;
    if (argc == 1) {
        printf("Error: missing argument\n");
        exit (0);
    }
    argv++;
    if (sscanf(*argv, "%d", &n) != 1 || n < 0 || n >= MAXNUM)
    {
        printf ("Error: %s not an int or out of range\n",
    *argv);
        exit (0);
    }
    printf("fact(%d) = %d\n", n, fact(n));
}

/* fact2.c */
int* table;
int fact(int n) {
    static int num = 2;
    if (n >= num) {
        int i = num;
        while (i <= n) {
            table[i] = table[i-1] * i;
            i++;
        }
```

```
        num = i;
    }
    return table[n];
}
```

（1）对于每个程序中的相应符号，给出它的属性（局部变量、强全局变量或弱全局变量），以及它在链接后位于 ELF 文件中的什么位置？（提示：如果某表项中的内容无法确定，请画 X）（6 分）

fact1.c

变量	类型	ELF Section
table	**weak global**	**.bss** COM
fact	**weak global**	**X（或.bss）**UND
num	**X**	**X**

fact2.c

变量	类型	ELF Section
table	**weak global**	**.bss** COM
fact	**strong global**	**.text**
num	**local**	**.data**

（2）对上述两个文件进行链接之后，会对每个符号进行解析。请给出链接后下列符号被定义的模块（fact1 or fact2）。（2 分）

	定义模块
table	**不确定**
fact	**fact2**
num	**fact2**

（3）使用 gcc（命令：gcc -o fact fact1.c fact2.c）来编译之后得到的可执行文件是否能够正确执行？为什么？（2 分）

不一定能正确执行。因为 table 在两个文件中都是 weak global，因此链接器会任选一个定义来解析 table。而因为在 fact2 模块中只给 table 预留了一个单字（4 字节）空间，因此如果选择了 fact2 来解析 table 的话，会出现 segmentation fault。【该问题已经过编译检查确认。】

考察知识点：linking 的基本概念、符号属性、链接的过程、符号的解析等。

一定不能正确执行。即使正确的选择fact1里的定义来解析table，fact2里的table指针也会将table数组的前8个字节当作自己所指向的地址值，引起错误。

得分

第五题（10 分）

Part I

请阅读以下程序，然后回答问题（假设程序中的函数调用都可以正确执行）：

```
int main() {
    printf("A\n");
    if (fork() == 0) {
        printf("B\n");
    }
    else {
        printf("C\n");
        A
    }
    printf("D\n");
    exit(0);
}
```

（1）如果程序中的 A 位置的代码为空，列出所有可能的输出结果：（1 分）

4 个：分别是 ABDCD ABCDD ACBDD ACDBD（错一个扣半分，多了也扣半分，最多扣 1 分）

（2）如果程序中的 A 位置的代码为：

```
waitpid(-1, NULL, 0);
```

列出所有可能的输出结果：（2 分）

3 个：分别是 ABDCD ABCDD ACBDD（每个半分，多了扣一分，最多扣 2 分）

（3）如果程序中的 A 位置的代码为：

```
printf("E\n");
```

列出所有可能的输出结果：（2 分）

7 个：分别是 ABDCED ABCEDD ACEBDD ACEDBD ACBEDD ACBDED ABCDED（错一个扣半分，多了扣一分）

Part II

请阅读以下程序，然后回答问题（假设程序中的函数调用都可以正确执行，且每条语句都是原子动作）：

```
pid_t pid;
int even = 0;
int counter1 = 0;
int counter2 = 1;
void handler1(int sig) {
    if (even % 2 == 0) {
        printf("%d\n",
    counter1);
        counter1 =     A     ;
        } else {
        printf("%d\n",
    counter2);
        counter2 =     B     ;
        }
        even =     C     ;
}
void handler2(int sig) {
        if (     D     ) {
        counter1 = even * even;
        } else {
        counter2 = even * even;
        }
}
int main() {
    signal(SIGUSR1, handler1);
    signal(SIGUSR2, handler2);
    if ((pid = fork()) == 0) {
        while (1) {};
        }
        while (even < 20) {
            kill(pid,    E    );
            sleep(1);
            kill(pid,    F    );
            sleep(1);
            even += 2;
        }
        kill(pid, SIGKILL);
        exit(0);
    }
```

1)．完成程序，使得程序在输出前 20 个斐波那契(Fibonacci)数列，即 $F_0=0$, $F_1=1$, …, $F_n=F_{n-1}+F_{n-2}$。（如果存在对本次程序执行结果没有影响的语句，请在相应位置填写“无关”）（3 分）

A: ___counter1 + counter2__

B: ___counter1 + counter2__

C: ___even + 1_____________

D: _____无关_____________

E: ___SIGUSR1_____________

F: ___SIGUSR1_____________

2)．完成程序，其中 A，B 处保持不变，使得程序可以分别输出前几个奇数或偶数的平方和。（如果存在对本次程序执行结果没有影响的语句，请在相应位置填写“无关”）（2 分）

其中：

若要输出奇数的平方和：even 的初始值为 **3**；

若要输出偶数的平方和，even 的初始值为 **2**。

C: _____**even + 2**_____________

D: _____**even % 2 == 1**_______

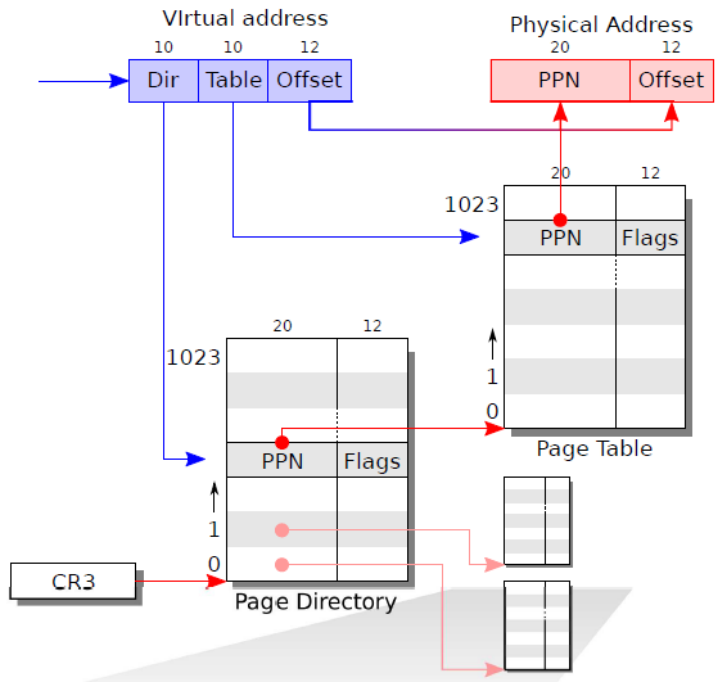
E: _____**SIGUSR2**_____________

F: _____**SIGUSR1**_____________

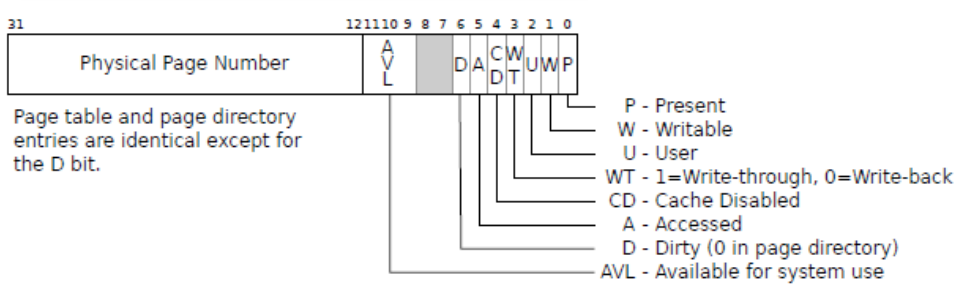
得分

第六题（10 分）

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)。其虚拟地址到物理地址的翻译方式如下图。先根据 CR3 找到页目录地址，然后依据偏移 Dir 找到一个页目录项，页目录项的高 20 位(PPN)为二级页表地址；在二级页表中根据偏移 Table 找到页表项，页表项中的高 20 位(PPN)即为物理地址的高 20 位，将这 20 位与虚拟地址的低 12 位拼在一起形成完整的物理地址。



页目录和页表均有 1024 项，每一项为 4 字节，含义如下：



页目录和页表由操作系统维护，通常情况下只能在内核态下访问，为了给用户提供一个访问页表项和页目录项内容的接口，假设操作系统中已经执行过如下代码段：

```
#define UVPT 0xef400000
#define PDX(la)  ((((unsigned int) (la)) >> 22) & 0x3FF)
……
kern_pgdir[PDX(UVPT)] = PADDR(kern_pgdir) | PTE_U | PTE_P;
```

其中 kern_pgdir 是操统系统维护的页目录数组，共 1024 项，每一项的类型为 unsigned int。PADDR(kern_pgdir)用于获得 kern_pgdir 的物理地址，页目录在物理内存中正好占一页，所以 kern_pgdir 的物理地址是 4KB 对齐的。PTE_U 和 PTE_P 代表了这个页目录项的权限，即用户态可访问(只读)。可以看到，这条语句将页目录的第 PDX(UVPT)项指向了页目录自身。

利用这一点，对于给定的虚拟地址 va，可以获得 va 对应的页目录项和页表项内容，分别对应于函数 get_pde 和 get_pte，请完成这两个函数(每空 2 分)：

```
#define UVPT 0xef400000
// get_pde(va): 获取虚拟地址 va 对应的一级页表(页目录)中的页目录项内容
unsigned int get_pde(unsigned int va) {
    unsigned int pdx = (va >> _[1]_____ ) & _[2]_____;
    unsigned int addr = UVPT + (_[3]_____) + pdx * 4;
                return *((unsigned int *)(addr));
}

// get_pte(va): 获取虚拟地址 va 对应的二级页表中的页表项内容
unsigned int get_pte(unsigned int va) {
    unsigned int PGNUM = va >> ___[4]_____;
    unsigned int addr = ___[5]_____ + PGNUM;
    return *((unsigned int *)(addr));
}
```

- 答案：

  [1]: 22　　[2] 0x3ff (或 1023)　[3] UVPT >> 10

  [4]: 10　　[5] UVPT

```
#define UVPT 0xef400000

unsigned int get_pde(unsigned int va) {
    unsigned int pdx = (va >> 22) & 0x3ff;
    unsigned int addr = UVPT + (UVPT >> 10) + pdx * 4;
    return *((unsigned int *)(addr));
}

unsigned int get_pte(unsigned int va) {
    unsigned int PGNUM = va >> 12;
    unsigned int addr = UVPT + PGNUM * 4;
    return *((unsigned int *)(addr));
}
```

要获取页目录项，需要让 addr= UVPT[31:22] | UVPT[31:22] | PDX | 00，即最高的 10 位为 UVPT 的高 10 位，中间的 10 位也是 UVPT 的高 10 位，接下来 10 位是 PDX(va 的高 10 位)，最后两位为 0。

这样，根据翻译机制，第一步取出 addr 的高 10 位 UVPT[31:22]，由于页目录中 UVPT[31:22]指向的是页目录本身，所以这一步找到的二级页表仍然为页目录本身。第二步取出中间 10 位 UVPT[31:22]，重复刚才的过程，这一步找到的物理页仍然为页目录本身。最后加上偏移量 PDX|00 就可以找到页目录中的第 PDX 项。

要获取页表项，让 addr= UVPT[31:22] | PDX | PTX | 00，其中 PTX 是 va 的中间 10 位。这样，第一步翻译后得到的二级页表仍然指向页目录本身，第二步翻译后得到的物理页指向的是 va 对应的二级页表。最后用偏移量 PTX|00 即可获得二级页表中的第 PTX 项。

得分

第七题（10 分）

（1） 一个服务器拥有两个独立的固定 IP 地址，那么它在 web 应用端口 80 上最多可以监听多少个独立的 socket 连接？（2 分）

答案：$2*2^{48}$

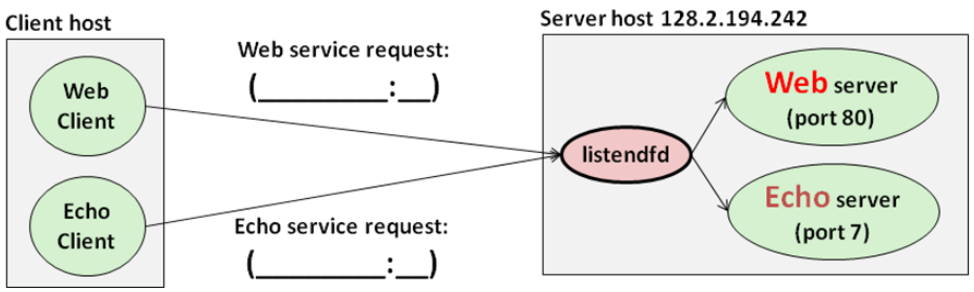
服务器端	客户端	结果
2 个独立固定 IP	任意 32 位 IP 任意 16 位 port number	$2*2^{32+16}$

（2）该服务器在所有有 web 应用端口上最多可以监听多少个独立的 socket 连接？（2 分）

答案：$2*2^{64}$

服务器端	客户端	结果
2 个独立固定 IP 任意 16 位 port number	任意 32 位 IP 任意 16 位 port number	$2*2^{16+32+16}$

（3） 在下图中连线上填入正确的目标服务器的 socket 标识符（2 分）



答案：

128.2.194.242:80

128.2.194.242:7

4) 在 Echo server 范例中，server 端通过 accept 函数接受了一个 client 的连接请求，从而将网络描述符与该网络连接、socket 绑定，然后进行网络数据传输。在下面的空格处填写正确的网络描述符，每个空填写 **listenfd** 或 **connfd**（共 4 分，每空 1 分）。

```
int main(int argc, char **argv) {
    int listenfd, connfd, port, clientlen;
    struct sockaddr_in clientaddr;
    struct hostent *hp;
    char *haddrp;
    unsigned short client_port;
       ……
       while (1) {
           clientlen = sizeof(clientaddr);
           __________ = Accept(__________, (SA *)&clientaddr,
                       &clientlen);
           hp = Gethostbyaddr((const char*)
                &clientaddr.sin_addr.s_addr,
                sizeof(clientaddr.sin_addr.s_addr), AF_INET);
           haddrp = inet_ntoa(clientaddr.sin_addr);
           client_port = ntohs(clientaddr.sin_port);
           printf("server connected to %s (%s), port %u\n",
                     hp->h_name, haddrp, client_port);
           echo(__________);
           Close(__________);
           }
       }
```

答案：connfd, listenfd, connfd, connfd

accept 之后把参数里的 listenfd 转交给 connfd

server 无法支持多个 connections，因为只有一个文件描述符（listenfd）

考察学生对 网络文件描述符、listenfd、connfd 的理解。

得分

第八题（10 分）

某个城市为了解决市中心交通拥堵的问题，决定出台一项交通管制措施，对进入市中心区的机动车辆实行单双日限制行驶的办法。具体要求是，逢单日，只允许车辆牌号号码为单数的机动车进入市中心区；同样，逢双日，只允许车辆牌号号码为双数的机动车进入市内中心区。有一个进入市中心区的交通路口，进入该路口的道路有一条，离开该路口的道路有两条，其中一条是通往市中心区的道路，而另一条是绕过市中心区的环路，在进入路口处设置了自动识别车辆牌号的识别设备与放行栅栏控制设备。在单日，遇到单号车辆进入路口车辆号码识别区，号码识别设备打开通往市中心区道路的放行栅栏；而遇到双号车辆，则打开绕过市中心区环路的放行栅栏。反之亦然。显然，只有在该路口车辆号码识别区中无车时，才允许一辆车进入车辆号码识别区。同时为了防止有车辆混过路口，两个放行栅栏平时处于关闭状态，只有在车辆号码识别区中的车辆已被识别出单双号之后，放行栅栏才会在识别设备的控制下，打开对应的放行栅栏，在车辆通过之后，该放行栅栏自行关闭。

```
vehicle_n  int;                    /* 车辆号码 */
```

检查车辆牌号线程 T1：

```
 while (1) {
    车辆到达识别区路口；
    ①
    车辆进入号码识别区；
    if  (vehicle_n == 奇数)
       {    ②        }
    else
       {    ③        }
  };
```

市区放行栅栏线程 T2：

```
 while (1) {
    ④
    允许车辆进入市中心区；
    ⑤
  };
```

环路放行栅栏线程 T3：

```
while (1) {
    ⑥
    允许车辆绕行环路；
    ⑦
};
```

1）请设计若干信号量，给出每一个信号量的作用和初值。（3 分）

参考答案：

Check：指示可否在车辆号码识别区中进入一辆汽车，由于只能进入一辆，其初值为 1。

Odd：指示汽车号码是否为奇数，其初值为 0，表示不是奇数。

Even：指示汽车号码是否为偶数，其初值为 0，表示不是偶数。

2）请将信号量上对应的 PV 操作填写在代码中适当位置。（7 分）

参考答案：

① P(Check)；

② V(Odd)；

③ V(Even)；

④ P(Odd)；

⑤ V(Check)；

⑥ P(Even)；

⑦ V(Check)；