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：2018 年 01 月 07 日 **小班编号：** ____ **小班教师:** _____

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

装订线内

不要答题

北京大学考场纪律

1、考生要按规定的考试时间提前 5 分钟进入考场，隔位就坐或按照监考人员的安排就座，将学生证放在桌面。无学生证者不能参加考试；迟到超过 15 分钟不得入场；与考试无关人员不得进入考场。考生在考试开始 30 分钟后方可交卷出场；未交卷擅自离开考场，不得重新进入考场继续答卷；交卷后应离开考场，不得在考场内逗留或在考场附近高声交谈。

2、除非开卷考试中教师另有说明，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、手机等）不得带入座位，已经带入考场的手机等电子设备必须关机，不得随身携带或放在座位旁边，应与其他物品一起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。若有试题印制问题应向监考人员提出，不得向其他考生询问。考生提前答完试卷，应举手示意请监考人员收卷后方可离开；考试结束监考人员宣布收卷时，考生应立即停止答卷，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。不准旁窥、交头接耳、打暗号或做手势，不准携带与考试内容相关的材料参加考试，不准使用手机、非教师允许的计算器等具有信息发送、接受、存储功能的设备，不准抄袭或协助他人抄袭试题答案或者与考试内容相关的资料，不准传、接或者交换试卷、答卷、草稿纸，不准由他人代替考试或替他人参加考试等。凡违反考试纪律或作弊者，按《北京大学本科考试工作与学习纪律管理规定》给予相应处分。

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。

以下为试题和答题纸，共 21 页。

得分

第一题 单项选择题（每小题 1 分，共 20 分）

注：选择题的回答填写在下表中。

题号	1	2	3	4	5	6	7	8	9	10
回答										
题号	11	12	13	14	15	16	17	18	19	20
回答										

1. 在 gdb 调试中，下一条指令是 call func1。下面哪条 gdb 指令能执行该指令并且停留在 func1 的第一条指令？

A. br func1　　B. si　　C. ni　　D. disas func1

2. 浮点数的阶码没有设计成补码的形式的原因是：

A. 为了加快四则运算　　B. 为了方便排序

C. 为了保存更多数值　　D. 为了表示特殊值

3. 下面说法正确的是：

A. 不同指令的机器码长度是相同的

B. test %rax, %rax 恒等于 cmp $0, %rax

C. switch 编译后总是会产生跳转表

D. 以上都不对

4. 分析下图的指令执行步骤，请问这是 Y86 指令系统的哪条指令？

Fetch	icode:ifun ← M_1[PC]
Decode	valA ← R[%esp] valB ← R[%esp]
Execute	valE ← valB + 4
Memory	valM ← M_4[valA]
Write back	R[%esp] ← valE
PC update	PC ← valM

A. call

B. ret

C. pushl

D. popl

5. 如果四路组相联的高速缓存大小是 32KB，并且块（block）大小为 32 字节，那么它每路（way）有多少行（line）？

A. 32
B. 64
C. 128
D. 256

6. 某个机械硬盘的部分指标数据如下，则该硬盘标称的容量最有可能是多少？
   - 1024 bytes/sector
   - 400 sectors/track (on average)
   - 20000 tracks/surface
   - 5 platters/disk

A. 80 GB
B. 76 GB
C. 40 GB
D. 38 GB

7. C 源文件 f1.c 和 f2.c 的代码分别如下所示：

```
// f1.c
#include <stdio.h>
void f();
int x ;
int main(void)
{
   x = 1;
   f() ;
  printf("%x\n", x) ;
  return 0;
}
```

```
//f2.c
float x ;
void f()
{
  x = 2 ;
}
```

运行下面的命令后得到的结果是：

```
$ gcc f1.c f2.c
$ ./a.out
```

A. 1　　B. 2　　C. 3f800000　　D. 40000000

8. 下面关于链接的说法，正确的是：

A. Linux 链接器在处理多重定义的同名弱符号时，选择链接时遇到的第一个符号
B. 链接发生在源代码编译之后、可执行目标程序运行之前
C. C 程序静态局部变量和静态全局变量都在 ELF 可重定位目标文件的.data 段
D. 链接器构造可执行目标文件时，只复制静态库里被应用程序引用的目标模块

9. 下列程序输出的数字顺序可能是：

```
int count = 1;
if (fork() == 0) {
    if (fork() == 0) {
        printf("%d\n", ++count);
    }
    else {
        printf("%d\n", --count);
    }
}
printf("%d\n", ++count);
```

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

10. 下列关于信号的说法不正确的是：
A. 在键盘上输入 Ctrl-C 会导致内核发送一个 SIGINT 信号到前台进程组中的每个进程
B. 每种类型最多只能有一个未处理的信号
C. SIGINT 的处理函数不能被另一个 SIGINT 信号中断
D. 进程可以通过使用 signal 函数修改和 SIGSTOP 相关联的默认行为

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

14. 以下关于文件 I/O 的说法中，正确的是：

A. 文件重定向（dup 和 dup2）操作仅仅改变了文件描述符的指向，不会改变打开文件表中的内容

B. 进程调用 fork()时，可能对文件描述符表和打开文件表采用写时拷贝（Copy on Write）机制

C. 对同一描述符，rio_readlineb 和 rio_readnb 可以任意交叉使用，rio_readn 和 rio_writen 也可以任意交叉使用

D. RIO 中包括无缓冲的输入输出函数和带缓冲的输入输出函数，使用带缓冲的输入输出函数时，要先声明一个 rio_t 类型变量并调用 rio_readinitb 函数

15. 以下程序执行完成后，ICS.txt 文件中的内容是：

```
int main(int argc, char** argv) {
    int fd1 = open("ICS.txt", O_CREAT|O_RDWR,
                                S_IRUSR|S_IWUSR);
    write(fd1, "ics ", 4);
    int fd2 = fd1;
    int fd3 = dup(fd2);
    int fd4 = open("ICS.txt", O_APPEND|O_RDWR);
    write(fd2, "segmentation fault ", 19);
    write(fd4, "tao", 3);
    int fd5 = fd4;
    dup2(fd3, fd5);
    write(fd4, "lab", 3);
    close(fd1);
    return 0;
}
```

A. ics segmentation fault tao

B. ics segmentation fault lab

C. ics taomentation fault lab

D. tao segmentation fault lab

16. 一台处于校园网内的笔记本电脑访问一台处于公网上的服务器上的网页服务。网页服务使用默认的 80 端口。以下答案有三项一定不正确，可能正确的那一项是：

A. 笔记本在通信过程中，本地使用的端口号是 80

B. 笔记本的 IP 地址是 192.168.1.101

C. 服务器向笔记本传送网页内容时，使用的服务器端口号是 80

D. 服务器的 IP 地址是 192.168.1.102

17. 下列关于计算机网络概念的说法中，正确的是：

A. HUB 与交换机都会转发从任意端口上收到的帧，区别在于交换机转发速度更快

B. UDP 是无连接的协议，而实时音视频一般需要先呼叫对方建立连接，因此这些应用不能基于 UDP 传输音视频数据

C. 处于数据链路层的以太网协议和处于传输层的 TCP 协议都对数据进行了校验

D. HTML 是互联网上应用最为广泛的超文本传输协议

18. 下列关于计算机网络的说法中，错误的是：

A. 在 Linux 系统中，每台主机都有本地定义的域名 localhost，这个域名总是被映射为 IP 地址 127.0.0.1

B. 在套接字编程中，既可以用网卡地址作为地址，也可以用 IP 地址作为地址

C. Web 客户端和服务器间的交互采用的是基于文本的无连接协议

D. IPv4 中的 IP 地址是一个 32 位无符号整数，IPv6 中的 IP 地址是一个 128 位无符号整数

19. 下列关于死锁的叙述中，不正确的是：

A. 所谓死锁是指一组线程被阻塞，等待一个永远不会为真的条件

B. 在用信号量及 PV 操作解决生产者消费者问题（多个缓冲区、多个生产者，多个消费者）时，如果先给缓冲区加锁 P(&sp->mutex)，再申请可用的缓冲区槽 P(&sp->slots)；则可能出现死锁

C. printf()是线程安全函数

D. 在信号处理函数中使用 printf()不会导致死锁

20. 某进程的主线程和对等线程的代码如下所示：

```
sem_t sem;

int main()
{
    int i;
    pthread_t tids[3];
    sem_init(&sem, 0, 1);

    for (i=0; i<3; i++) {
        pthread_create(&tids[i], NULL, thread, NULL);
    }

    for (i=0; i<3; i++) {
        pthread_join(&tids[i], NULL);
    }
    return 0;
}

int y = 15;
void *thread (void *arg)
{
    P(&sem);
    y = y - 3;
    V(&sem);
    printf("%d\n", y);
}
```

执行上述代码后，会产生多少种不同的输出？

A. 11　　B. 12　　C. 13　　D. 14

得分

第二题（12 分）

假设程序运行在 x86-64 的小端法机器上，编译器按照标准规则对齐，并且有 nan 参与的算术比较运算都返回 0。请回答下列问题：

1．请根据左边 C 程序和右边对应的汇编代码，填写空白处的代码。

```
struct a {
    char c[2][3];
    int i;
};
struct b {
    float f;
    struct a a;
    char c;
};
union c {
    struct a a;
    struct b b;
    struct b* pb;
};
void proc1(union c* u){
    double a = u->b.f_____;
    double b = u->b.f_____;
    printf("%d\n", a==b);
}

void proc2(union c* u){
    char* a = u->a.c[1] + 1;
    char* b = (u->a.c + 1)[1];
    printf("%d\n", b - a);
}

void proc3(union c* u){
    char x = u->a.c[1][0];
```

```
……
proc1:
        ……
        movss     (%rcx), %xmm1
        movss     .LC0(%rip), %xmm0
        addss     %xmm1, %xmm0
        cvtss2sd  %xmm0, %xmm2
        movsd     %xmm2, -8(%rbp)
        movss     (%rcx), %xmm0
        cvtss2sd  %xmm0, %xmm0
        movsd     .LC1(%rip), %xmm1
        addsd     %xmm1, %xmm0
        movq      %xmm0, -16(%rbp)
        ……
proc2:
        ……
proc3:
        ……
        _____________________________
        movb      %al, -1(%rbp)
        movzbl    2(%rcx), %eax
        movb      %al, -2(%rbp)
        ……
proc4:
        ……
        movl      8(%rcx), %eax
        movl      %eax, -4(%rbp)
        movzbl    16(%rcx), %eax
```

```
    char y = ____________________;
    if (x == 0x7F && (y & 0x80) == 0x80)
        printf("%d\n", (u->b.f < u->a.i));
    else
        printf("0\n");
}

void proc4(union c* u){
    int x = u->a.i;
    int y = _______________;
    printf("%d\n", (x > y) != (-x<-y));
}

void main(){
    unsigned int i;
    printf("%d\n", sizeof(struct a));
    printf("%d\n", sizeof(struct b));
    printf("%d\n", sizeof(union c));

    union c u;
    …… //随机初始化 u
    proc1(&u);
    proc2(&u);
    proc3(&u);
    proc4(&u);
}
```

```
        movsbl    %al, %eax
        movl      %eax, -8(%rbp)
        ……
main:
        ……
        leaq 32(%rsp), %rbx
        movq      %rbx, %rcx
        call  proc1
        movq      %rbx, %rcx
        call  proc2
        movq      %rbx, %rcx
        call  proc3
        movq      %rbx, %rcx
        call  proc4
        ……
        .align 4
.LC0:
        .long     1048576000
                  //十六进制 3E800000
        .align 8
.LC1:
        .long     0
        .long     1070596096
                  //十六进制  3FD00000
```

2．程序执行会产生 7 行输出。请写出每一行所有可能的输出值，不同可能性之间逗号分隔

第一行：_____

第二行：_____

第三行：_____

第四行：_____

第五行：_____

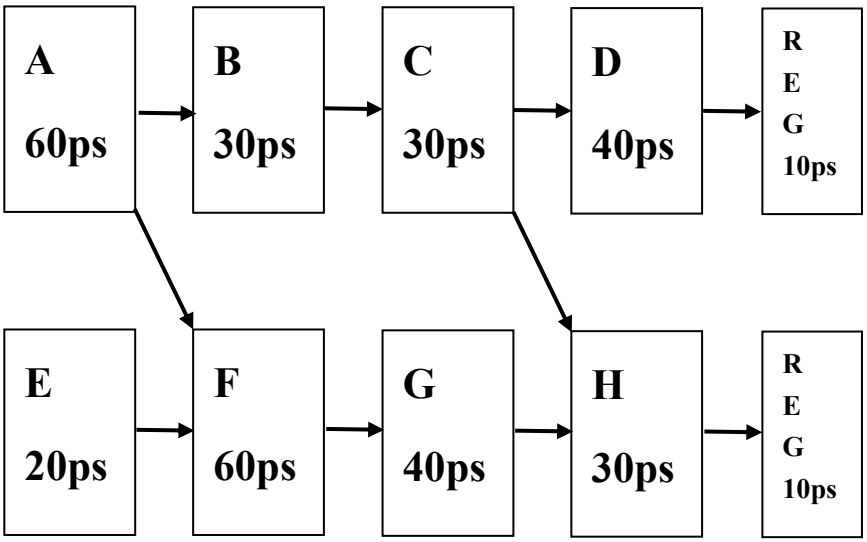
第六行：_____

第七行：_____

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，每个单元的延迟以及数据依赖关系已在图中标出。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 10ps。



1．如果改造为一个二级流水线，为获得最大的吞吐率：

a）需要在_____模块之间和_____模块之间插入寄存器；

b）插入寄存器改造后二级流水线的最长延时路径为_____，最长延时为____

c）流水线最大吞吐率为：_____

（用分数表示，约分到最简形式，并写明单位）

2．如果改造为一个三级流水线，为获得最大的吞吐率：

a）需要在_____

模块之间插入寄存器；

b）插入寄存器改造后三级流水线的最长路径延时为:_____；

c）流水线最大吞吐率为：_____

（用分数表示，约分到最简形式，并写明单位）

得分

第四题（10 分）

考虑以下三个文件：

polygon.h

```
struct Node {
    float pos[2];
    int marked;
    struct Node* next;
    struct Node* prev;
};
typedef struct Node node;

node* alloc();
void init();
void gc();
```

main.c（函数中部分内容折叠）

```
#include "polygon.h"
node* root_ptr ;
int main(){
    node* p;
    init();
    p=alloc();
    root_ptr =p;
    ...
    gc();
    ...
    return 0;
}
```

gc.c（函数体被折叠）

```
#include "polygon.h"
#define N (1<<20)
static node polygon [N];
static node* free_ptr ;
static node* root_ptr ;
void mark(node* v){...}
void sweep() {...}
void gc() {...}
void init() {...}
node* alloc() {...}
```

使用命令 gcc –o polygon main.c gc.c 得到可执行文件 polygon。

1．对于每个程序中的相应符号，给出它的属性（局部或全局，强符号或弱符号）
提示：如果某表项中的内容无法确定，请画 X。

main.c

	局部或全局?	强或弱?
root_ptr		
init		
main		

gc.c

	局部或全局?	强或弱?
N		
polygon		
alloc		

2．解释为何其中一些符号被定义了多次，链接器仍然可以成功创建可执行文件。

3．gc.c 的功能是实现一个垃圾收集器。解释为何前面的命令能够编译、链接成功，但得到的执行文件中却存在潜在错误。并试提出如何修复这一 bug。

得分

第五题（10 分）

代码如下：（仅为示例代码，答题时不考虑任何出错的情况）

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/wait.h>

int counter1 = 1;
int counter2 = 1;

void handler(int sig) {
       if (sig == SIGUSR1)    counter1++;
       if (sig == SIGUSR2)    counter2++;
}

int main(int argc, char **argv)
{
       int i;
       pid_t ppid, cpid;
       int fd1, fd2;
       unsigned char buf1[64], buf2[64];

       signal(SIGUSR1, handler);
       signal(SIGUSR2, handler);

       fd1 = open("number.txt", O_RDWR);

       for (i = 0; i < __N__; i++) {
              ppid = getpid();
              cpid = fork();
              if (cpid) {
```

```
                kill(cpid, ___A_____ );
                while (sleep(4)) ;
        } else {
                kill(ppid, ___B_____ );
                while (sleep(2)) ;
        }
        fd2 = open("letter.txt", O_RDWR);
        read(fd1, buf1, counter1);
        read(fd2, buf2, counter2);
        buf1[counter1] = '\0';
        buf2[counter2] = '\0';
        printf("%s %s\n", buf1, buf2);
        if (cpid) {
                waitpid(cpid, 0, 0);
        } else {
                _____E_____;
        }
    }
}
```

其中，number.txt 文件内容为：

12345678901234567890123456789O

其中，letter.txt 文件内容为：

abcdefghijklmnopqrstuvwxyz

1．当 N=2，A/B 为 SIGUSR1 或 SIGUSR2，E 为空时，共有_____行输出

a） A 为 SIGUSR1，B 为 SIGUSR1 时，最后一行输出为_____

b） A 为 SIGUSR1，B 为 SIGUSR2 时，最后一行输出为_____

c） A 为 SIGUSR2，B 为 SIGUSR1 时，最后一行输出为_____

d） A 为 SIGUSR2，B 为 SIGUSR2 时，最后一行输出为_____

2．当 N=3 时，A/B 为 SIGUSR1 或 SIGUSR2，E 为 exit(0)时，共有_____行输出

a） A 为 SIGUSR1，B 为 SIGUSR1 时，最后一行输出为_____

b） A 为 SIGUSR1，B 为 SIGUSR2 时，最后一行输出为_____

c） A 为 SIGUSR2，B 为 SIGUSR1 时，最后一行输出为_____

d） A 为 SIGUSR2，B 为 SIGUSR2 时，最后一行输出为_____

得分

第六题（12 分）

通常处理器中的 MMU 通过页表实现虚拟内存地址到物理内存地址的转换，而 TLB 被用于提升地址转换的效率。然而 TLB 必须和页表之间保持一致，才能保证 MMU 正确按页表转换虚拟地址。当修改页表内容时，需要通过 invlpg 指令将对应的 TLB 表项置为失效，以确保 TLB 与页表一致。

指令 invlpg m 把包含虚拟地址 m 的页面在 TLB 中的表项置为失效。

假设当前状态下，TLB 与页表是一致的。部分 TLB 表项如下（假设 TLB 是全相联的）：

有效位	TLB 标记	页面号
1	0x8040201	0x4801
1	0x8040233	0x4812
0	0x8040382	0x9C33
1	0x8046740	0x4801
0	0x8046621	0x8845
1	0x80467CD	0x6734

与上述 TLB 表项相关的两个页表页的地址及其中的部分页表项如下：

索引号	页面号	存在位
	Bits: 51 - 12	Bit: 0
0x1	0x4801	1
0x33	0x4812	1
0x54	0x8745	1
0x180	0x3212	1
0x182	0x9C33	1
0x1A1	0x9078	1

基地址为 0x4801000 的页表页

索引号	页面号	存在位
	Bits: 51 - 12	Bit: 0
0x10	0x2312	1
0x21	0x8845	1
0x73	0x4521	1
0x140	0x4801	1
0x182	0x8ACD	1
0x1CD	0x6734	1

基地址为 0x4812000 的页表页

设处理器采用的是 64 位虚拟地址和 64 位物理地址，页面大小为 4KB，页表为 4 级。

1．分析下面的指令序列：

```
1. movq $0x8040233000, %rbx
2. movq 0xA00(%rbx), %rcx
3. addq 0x11000, %rcx
4. movq %rcx, 0x108(rbx)
5. movq 0x8046621108, %rax
```

a) 执行完第 2 行指令后，载入到%rcx 中的物理页面号为:_____；

b) 执行完第 3 行指令后，%rcx 中的物理页面号为：_____；

c) 执行完第 5 行指令后，%rcx 和%rax 中的内容之间有何关系？_____

d) 在执行上述 5 行指令过程中，会发生_____次 TLB miss;

e) 在执行上述 5 行指令过程中，会发生_____次 page fault;

2．接着上面的的 5 行指令接着执行下面的指令:

```
6. movq 0x8040382C10, %r10
7. movq %rax, 0x8046740C10
8. movq 0x8040382C10, %r11
9. invlpg 0x8040382C10
10. movq 0x8040382C10, %r12
```

a) 全部 10 条指令执行后，%r10 和%r11 中的内容之间有何关系？_____

b) 全部 10 条指令执行后，%r11 和%r12 中的内容之间有何关系？_____

c) 在执行上述 10 行指令过程中，共发生了_____次 TLB miss.

3．请写出执行完上述 10 行指令后，TLB 和页表页中有变化的表项中的内容。

有效位	TLB 标记	页面号
	0x8040201	
	0x8040233	
	0x8040382	
	0x8046740	
	0x8046621	
	0x80467CD	

索引号	页面号	存在位
	Bits: 51 - 12	Bit: 0
0x1		
0x33		
0x54		
0x180		
0x182		
0x1A1		

基地址为 0x4801000 的页表页

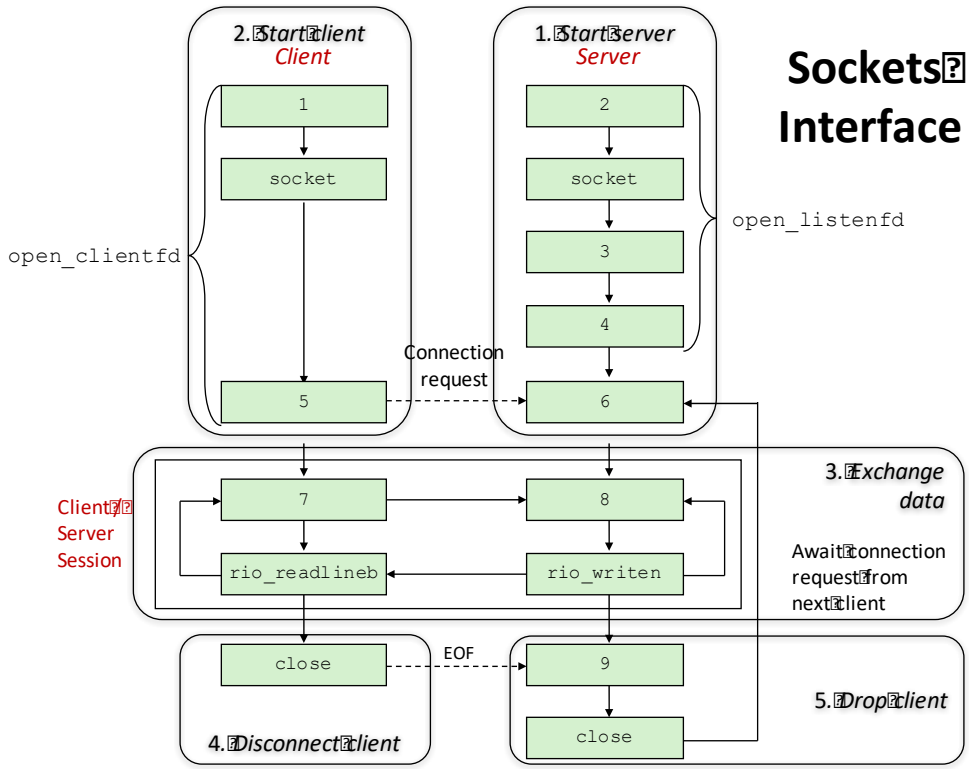
索引号	页面号	存在位
	Bits: 51 - 12	Bit: 0
0x10		
0x21		
0x73		
0x140		
0x182		
0x1CD		

基地址为 0x4812000 的页表页

得分

第七题（12 分）

1．请根据 Echo server/client 应用的标准程序填空。



1．_____ 2．_____ 3．_____

4．_____ 5．_____ 6．_____

7．_____ 8．_____ 9．_____

2．不定项选择题（全对得分，有错得 0 分）

A．TCP 是一种实现在用户态的端到端可靠数据传输协议

B．每一个连接在网络上的主机都会被分配一个固定的 IP 地址

C．IP 包在网际互联时通过其帧头（LAN frame header）和包头（Internet packet header）信息共同决定路由

D．HTTP1.0 支持在同一个连接里进行多个 transaction

E．Web 代理作为网络缓存功能的时候对用户是显式存在的

F．10.0.0.0/12 和 172.16.0.0/16 和 192.168.0.0/20 属于私有地址字段

上述正确的有：_____

得分

第八题（12 分）

1. 基于进程的并发 Echo 服务器代码（简称 P 代码）和基于线程的并发 Echo 服务器代码（简称 T 代码）如下所示：

P 代码：

```
#include "csapp.h"
void echo(int cnnfd);

void sigchld_handler(int sig)
{
    if (waitpid(-1, 0, WNOHANG) > 0)
           ;
    return;  }

int main(int argc, char **argv)
{  int listenfd, connfd;
   socklen_t clientlen;
   struct sockaddr_storage clientaddr;

   listenfd = Open_listenfd(argv[1]);
   while (1) {
      clientlen = sizeof(struct sockaddr_storage);
      connfd = Accept(listenfd,(SA*)&clientaddr,&clientlen);
      if (Fork() == 0) {
         echo(connfd);
         Close(connfd);
         exit(0);
      }
      Close(connfd);
   }
}
```

T代码：

```
#include “csapp.h”
void echo(int cnnfd);
void *thread(void *vargp);

int main(int argc, char **argv)
{  int listenfd, connfd1;
   socklen_t clientlen;
   struct sockaddr_storage clientaddr;
   pthread_t tid;

   listenfd = Open_listenfd(argv[1]);
   while (1) {
      clientlen = sizeof(struct sockaddr_storage);
      connfd1 = Accept(listenfd,(SA*)&clientaddr, &clientlen);
      Pthread_create(&tid, NULL, thread, connfd1);
   }
}

void *thread(void *vargp)
{
    int connfd2 = *((int *)vargp);
    Pthread_detach(pthread_self());
    echo(connfd2);
    Close(connfd2);
    return NULL;
}
```

问：以上 P 代码和 T 代码中存在几处错误，请指出出错的位置（请按照①、②等符号标注位置），解释出错的原因并给出修正错误的解决方案。

2. 假设有 3 个线程，代码如下所示。已声明了三个信号量 s1、s2 和 s3，其初值分别为 1、0、0。请将信号量 S1、S2、S3 对应的 P()和 V()操作填写在三个线程中的标号处（每空至多填写 1 个操作），要求三个线程并发执行的结果只能是 x = 10。

```
x = 1;

void thread1()
{
    ① _____
    x = x * 5;
    ② _____
}

void thread2()
{
    ③ _____
    x = x + 3;
    ④ _____
}

void thread3()
{
```