1. 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

2. 在某一 64 位体系结构中，每页的大小为 4KB，采用的是三级页表，每张页表占据 1 页，页表项长度为 8 字节。则虚拟地址的位数为_____Bit。如果要映射满 64 位的虚拟地址空间，可通过增加页表级数来解决，那么至少要增加到_____级页表。这个体系结构支持多种页大小，最小的三个页大小分别是 4KB、_____MB、_____GB

3. 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

4. 有如下程序内容:

```
#include "csapp.h"
const char* hello = "No use\n";
char* bye = NULL;
int fd1 = 1;
int fd2;

int main() {
    fd1 = open("hello.txt", O_RDWR);
    fd2 = open("bye.txt", O_RDWR);
    hello = mmap(NULL, 16, PROT_READ, MAP_SHARED, fd1, 0);
    bye = mmap(NULL, 16, PROT_READ | PROT_WRITE,
                MAP_SHARED, fd2, 0);
    for (int i = 0; i < 8; i++)
        bye[i] = toupper(hello[i]);
    /************A************/
    munmap(hello, 16);
    munmap(bye, 16);
    return 0;
}
```

(1)　将以下符号归属到各个 section 中:

symbol	text	data	bss	不是符号
main				
hello				
bye				
fd1				
fd2				
i				

(2) 代码运行到 A 处的时候，/proc/2333/maps 中的内容如下：

```
ADDRESS                     PERM   PATH
555555555000-555555556000   (1)    /home/eugen/vm/vm
555555556000-555555557000   r--p   /home/eugen/vm/vm
555555558000-555555559000   rw-p   /home/eugen/vm/vm
7ffff7dea000-7ffff7f62000   r-xp   /usr/lib/x86_64-linux-gnu/libc-2.31.so
7ffff7f62000-7ffff7fb4000   r--p   /usr/lib/x86_64-linux-gnu/libc-2.31.so
7ffff7fb4000-7ffff7fb6000   rw-p   /usr/lib/x86_64-linux-gnu/libc-2.31.so
7ffff7fc9000-7ffff7fca000   (2)    /home/eugen/vm/bye.txt
7ffff7fca000-7ffff7fce000   r--p   [vvar]
7ffff7fce000-7ffff7fcf000   r-xp   [vdso]
7ffff7fcf000-7ffff7fd0000   r--p   /usr/lib/x86_64-linux-gnu/(4)-2.31.so
7ffff7fd0000-7ffff7ff3000   r-xp   /usr/lib/x86_64-linux-gnu/(4)-2.31.so
7ffff7ff3000-7ffff7ffb000   r--p   /usr/lib/x86_64-linux-gnu/(4)-2.31.so
7ffff7ffb000-7ffff7ffc000   (3)    /home/eugen/vm/hello.txt
7ffff7ffc000-7ffff7ffd000   r--p   /usr/lib/x86_64-linux-gnu/(4)-2.31.so
7ffff7ffd000-7ffff7ffe000   rw-p   /usr/lib/x86_64-linux-gnu/(4)-2.31.so
7fffffffe000-7ffffffff000   rw-p   [(5)]
```

PERM 有四位。前三位是 r=readable，w=writeable，x=executable，如果是-表示没有这一权限。第四位是 s=shared，p=private，表示映射是共享的还是私有的。填写(1)-(5)的内容：

7ffff7ffd000 对应的页在页表中被标为了只读。对该页进行写操作会导致 Page Fault 吗？ 会导致该进程收到 SIGSEGV 信号吗？

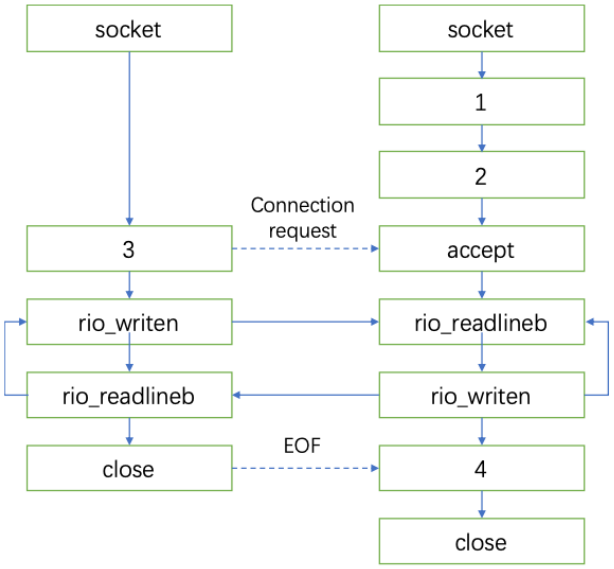
【TCP/IP 协议】请根据 web 应用在计算机网络中的定义以及其在协议栈自上而下在软件中的实现，把以下关键字填入表格

注：同一个关键词可能被填入多次；不是每一个关键词都必须被填入

Streams (end to end), Datagrams, web content, IP, TCP, UDP, Kernel code, User code

协议	数据包类型	软件实现
HTTP		

【套接字编程】补充下面的空格



【套接字编程】判断下列说法的正确性

2018 期末

( ) 套接字接口常常被用来创建网络应用

( ) Windows 10 系统没有实现套接字接口

( ) getaddrinfo()和 getnameinfo()可以被用于编写独立于特定版本的 IP 协议的程序

( ) socket()函数返回的描述符，可以使用标准 Unix I/O 函数进行读

( ) 数字数据只能通过数字信号传输

2015 期末

( ) 在 client-server 模型中，server 通常使用监听套接字 listenfd 和多个 client 同时通信

( ) 在 client-server 模型中，套接字是一种文件标识符

( ) 准确地说，IP 地址是用于标识主机的 adapter(network interface card)，并非主机

( ) Web 是一种互联网协议

( ) 域名和 IP 地址是一一对应的

( ) Internet 是一种 internet

2022 期末

( ) 对于 TCP 连接而言，其可以由连接双方的套接字对四元组（双方的 IP 地址和端口号）唯一确定。

( ) connect 函数用于客户端建立与服务器的连接，其参数中的套接字地址结构需要指明本客户端使用的端口号

( ) socket 函数返回的套接字描述符，可以立即使用标准 Unix I/O 函数进行读写

( ) bind 函数用于将用于客户端的主动套接字转化为用于服务器的监听套接字

( ) Web 服务使用 URL 来标明资源，这提供了一层抽象，使得客户端仿佛在访问远端的目录，而服务器处理了 URL 资源和具体文件/动态内容的映射关系

( ) Web 服务使用客户端-服务器模型，使用了 HTTP 协议，可以传输文本，HTML 页面，二进制文件等多种内容

( ) HTTP 响应会返回状态码，它指示了对响应的处理状态

【Web】使用浏览器打开网页 www.pku.edu.cn 的过程中，下列网络协议中，可能会被用到的网络协议有___个

HTTP TCP DNS IP

【Web】假设有一个 HTTPS(基于 HTTP 的一种安全的应用层协议)客户端程序想要 通过一个 URL 连接一个电子商务网络服务器获取一个文件，并且这个服务器 URL 的 IP 地址是已知的，以下哪种协议是一定不需要的?

A. HTTP B. TCP C. DNS D. SSL/TLS

【套接字】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 上最多能监听多少个独立的 socket 连接？该服务器在所有 web 应用端口上最多可以监听多少个独立的 socket 连接？

(不考虑特殊 IP 地址以及知名端口套接字)

【代理】

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, 208.216.181.15:80)	(155.232.108.39:51213, 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)