Homework 8

Problem 1

```
int main(){
    int fd1, fd2, fd3;
    char *buf1=(char*)malloc(10);
    char *buf2=(char*)malloc(10);
    fd1 = open("a.txt", O_RDWR, 0);
    fd2 = open("b.txt", O_RDWR|O_APPEND, 0);
    fd3 = open("a.txt", O_RDWR, 0);
    if(fork()==0){
        read(fd2, buf1, 2);
        dup2(fd1, fd2);
        read(fd2, buf1, 1);
        exit(0);
    }
    waitpid(-1, NULL, 0);
    read(fd2, buf1, 3);
    write(fd1, buf1, 3);
    read(fd1, buf1, 10);
    printf("%s\n", buf1);
    read(fd3, buf2, 10);
    dup2(fd2, 1);
    printf("%s\n", buf2);
    free(buf1);
    free(buf2);
    exit(0);
}
```

a. txt	abcdefg
b. txt	0123456789

1. What will the contents of a.txt and b.txt be after the program completes?

   a.txt: a234efg

   b.txt: 0123456789a234efg

2. What will be printed on stdout?

   efg

Problem 2

1. Please give three implementations of the following function, one uses *getnameinfo*, one uses *inet_ntop, ntohs* and one uses only *ntohs*:

    void print_sin(const struct sockaddr_in *addr);

    given an IPv4 address struct, print it as "x.x.x.x:port"

    (e.g. 127.0.0.1:1234)

    Note: For simplicity, you do NOT need to take care of error handling.

```
void print_sin1(const struct sockaddr_in *addr) {
    char host[100], port[100];
    getnameinfo(addr, sizeof(*addr), host, sizeof(host),
        port, sizeof(port), NI_NUMERICHOST |
        NI_NUMERICSERV);
    printf("%s:%s\n", host, port);
}

void print_sin2(const struct sockaddr_in *addr) {
    char host[100];
    inet_ntop(addr->sin_family, &addr->sin_addr, host,
        sizeof(host));
    printf("%s:%u\n", host, ntohs(addr->sin_port));
}

void print_sin3(const struct sockaddr_in *addr) {
    unsigned char *p = (unsigned char *) &addr->sin_addr;
    printf("%u.%u.%u.%u:%u\n", p[0], p[1], p[2], p[3],
        ntohs(addr->sin_port));
}
```

2. Assume we initialize addr as following:

```
struct sockaddr_in addr;
memset(&addr, 0, sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = 0x13784293;
addr.sin_port = 12387;
```

What's the output of print_sin? Why the port number is not 12387?

147.66.120.19:25392

Because the assign will store 12387 to sin_port in little-endian order, while socket related functions parse it as big-endian.

12387 = 0x3063 = 63 30 (stored in little-endian) = 0x6330 (loaded in big-endian) = 25392