CS5243 Advanced UNIX Programming
Assignment 12 (4 pts)
Group 4

(1) Screenshot of the code

```
// Initial Variables
char *buf = NULL;

while (fsz < sbuf.st_size)
{
    if ((sbuf.st_size - fsz) > COPYINCR)
        copysz = COPYINCR;
    else
        copysz = sbuf.st_size - fsz;

    /* TODO: Copy the file using mmap here */
    buf = (char *)mmap(NULL, copysz, PROT_WRITE, MAP_SHARED, fdout, fsz);

    // Close Output File Descriptor
    close(fdout);

    // Write memory
    read(fdin, buf, copysz);
    fsz += copysz;
}

// Close umap memory

munmap(buf, sbuf.st_size);
close(fdin);
exit(0);
```

(2) Screenshot of the result

```
freebsd@generic:~/Advanced-UNIX-Programming_Student/assignment12 % ./assignment12 source.txt dest.txt
freebsd@generic:~/Advanced-UNIX-Programming_Student/assignment12 % diff source.txt dest.txt
freebsd@generic:~/Advanced-UNIX-Programming_Student/assignment12 %
```

(3) Explanation

1. After calling **mmap**, we call **close**(fdout) to close the output file descriptor. However, the program works well, since closing the file descriptor doesn't unmap the region.
2. The **mmap** function returns the pointer of the mapped memory region. Next, we use **read** function to read the specified length content from the input file and store the data in the buffer, which is the mapped memory. Then, **munmap** close the memory

region. The result shows that the content of the source file and the destination file are the same.