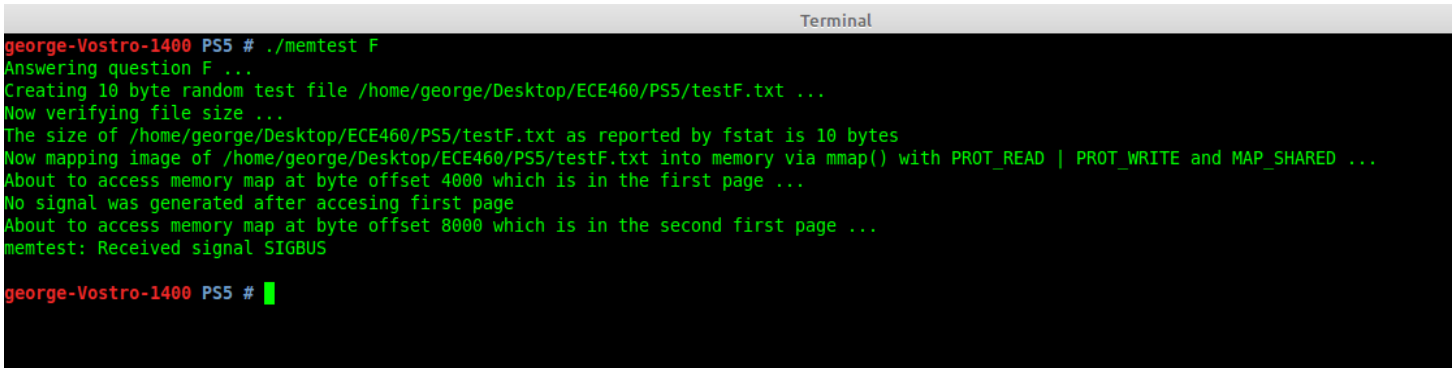
Stanley George
Professor Jeff Hakner
ECE 460: Operating Systems
Problem Set 5: Memory/mmap Test Programs

```
Terminal
george-Vostro-1400 PS5 # ./memtest A
Answering question A ...
Creating 64 byte random test file /home/george/Desktop/ECE460/PS5/testA.txt ...
The size of /home/george/Desktop/ECE460/PS5/testA.txt as reported by fstat is 64 bytes
Now mapping image of /home/george/Desktop/ECE460/PS5/testA.txt into memory via mmap() with PROT_READ and MAP_PRIVATE ...
Now writing value 0x01 at first byte of mapped memory ...
memtest: Received signal SIGSEGV

george-Vostro-1400 PS5 # ./memtest B
Answering question B ...
Creating 64 byte random test file /home/george/Desktop/ECE460/PS5/testBC.txt ...
The size of /home/george/Desktop/ECE460/PS5/testBC.txt as reported by fstat is 64 bytes
Now mapping image of /home/george/Desktop/ECE460/PS5/testBC.txt into memory via mmap() with PROT_READ | PROT_WRITE and MAP_SHARED ...
Now writing 64 bytes to mapped memory region starting at offset 0 ...
Now reading from file /home/george/Desktop/ECE460/PS5/testBC.txt into buffer of 64 bytes ...
In response to question B, when a file is mapped with MAP_SHARED and then writes to the mapped memory,
  the update IS visible when accessing the file through the read() system call

george-Vostro-1400 PS5 # ./memtest C
Answering question C ...
Creating 64 byte random test file /home/george/Desktop/ECE460/PS5/testBC.txt ...
The size of /home/george/Desktop/ECE460/PS5/testBC.txt as reported by fstat is 64 bytes
Now mapping image of /home/george/Desktop/ECE460/PS5/testBC.txt into memory via mmap() with PROT_READ | PROT_WRITE and MAP_PRIVATE ...
Now writing 64 bytes to mapped memory region starting at offset 0 ...
Now reading from file /home/george/Desktop/ECE460/PS5/testBC.txt into buffer of 64 bytes ...
In response to question C, when a file is mapped with MAP_PRIVATE and then writes to the mapped memory,
  the update IS NOT visible when accessing the file through the read() system call

george-Vostro-1400 PS5 #
```

```
Terminal
george-Vostro-1400 PS5 # ./memtest D
Answering question D ...
Creating 8195 byte random test file /home/george/Desktop/ECE460/PS5/testDE.txt ...
The initial size of /home/george/Desktop/ECE460/PS5/testDE.txt as reported by fstat is 8195 bytes
Now mapping image of /home/george/Desktop/ECE460/PS5/testDE.txt into memory via mmap() with PROT_READ | PROT_WRITE and MAP_SHARED ...
Now writing 4 bytes to memory map starting at byte offset 8195 ...
The new size of /home/george/Desktop/ECE460/PS5/testDE.txt as reported by fstat is 8195 bytes
File dump starting at offset 8195:
<EOF>
Memory dump starting at offset 8195 for 4 bytes:
<0x01> <0x02> <0x03> <0x04>
In response to question D, the file HAS NOT changed size
Now expanding /home/george/Desktop/ECE460/PS5/testDE.txt by 13 bytes and write(2) to new end of file ...
File dump starting at offset 8195:
<0x01> <0x02> <0x03> <0x04> <0x00> <0x00> <0x00> <0x00> <0x00> <0x00> <0x00> <0x41> <EOF>
Memory dump starting at offset 8195 for 16 bytes:
<0x01> <0x02> <0x03> <0x04> <0x00> <0x00> <0x00> <0x00> <0x00> <0x00> <0x00> <0x41> <0x00> <0x00> <0x00>
In response to question E, the data previously written to the hole ARE visible

george-Vostro-1400 PS5 # ./memtest E
Answering question E ...
First answering question D ...
Creating 8195 byte random test file /home/george/Desktop/ECE460/PS5/testDE.txt ...
The initial size of /home/george/Desktop/ECE460/PS5/testDE.txt as reported by fstat is 8195 bytes
Now mapping image of /home/george/Desktop/ECE460/PS5/testDE.txt into memory via mmap() with PROT_READ | PROT_WRITE and MAP_SHARED ...
Now writing 4 bytes to memory map starting at byte offset 8195 ...
The new size of /home/george/Desktop/ECE460/PS5/testDE.txt as reported by fstat is 8195 bytes
File dump starting at offset 8195:
<EOF>
Memory dump starting at offset 8195 for 4 bytes:
<0x01> <0x02> <0x03> <0x04>
In response to question D, the file HAS NOT changed size
Now expanding /home/george/Desktop/ECE460/PS5/testDE.txt by 13 bytes and write(2) to new end of file ...
File dump starting at offset 8195:
<0x01> <0x02> <0x03> <0x04> <0x00> <0x00> <0x00> <0x00> <0x00> <0x00> <0x00> <0x41> <EOF>
Memory dump starting at offset 8195 for 16 bytes:
<0x01> <0x02> <0x03> <0x04> <0x00> <0x00> <0x00> <0x00> <0x00> <0x00> <0x00> <0x41> <0x00> <0x00> <0x00>
In response to question E, the data previously written to the hole ARE visible

george-Vostro-1400 PS5 #
```



For better clarity, the above output was concatenated all into one file called memtest_out using a bash file and is attached to this document in one of the following pages

Comments for D

A write to mapped memory beyond the byte corresponding to the last byte of the file WILL NOT result in a change of the file size. This is probably because mmap only maps the bytes in the file up to the length specified by the length argument and zeros the rest of the bytes after the last byte in the file. The bytes which are 0 do not link back to the file, only those bytes which are a copy of the file link back to the file via MAP_SHARED. Thus, any write past the last byte of the file do not map back to the file or anything else. They will however reside in the mapped memory region.

Comments for F

An attempt to access memory that is within the mapped region but that is also beyond the page boundary of the file cause a SIGBUS signal to be delivered. However, if an access is attempted in the mmapped region that is also within the page boundary of the file, no signal will be delivered. This is because when an access outside the file page boundary occurs, the kernel is unable to resolve the page fault since no backing store exists. A page fault can be satisfied if the access is within the page boundary.