6. What is a user ID (UID)? Why is UID=0 significant? What is a group ID (GID)?
   **Reference:** *Introduction to Linux* (http://www.tldp.org/guides.html)

7. What is system call 24? What is system call 47? What parameters (if any) do each take? Where are the return values (if any) stored?

8. Consider the following two programs:

   ```assembly
   .section .bss
   .lcomm buffer, 32000
   .section .text
   .globl _start
   _start:
       movl $1, %eax
       movl $0, %ebx
       int $0x80
   
   and
   
   .section .data
   buffer:
       .fill 32000
   .section .text
   .globl _start
   _start:
       movl $1, %eax
       movl $0, %ebx
       int $0x80
   ```

   Compile each, and compare the size of each program. Explain any differences you see, or explain why there should not be any difference.

9. Consider the program `indexed.s` presented in class. Modify only the text section so that only every second prime number is moved into `eax`. 

---

COSC 647
Application Software Security
Assignment Sheet #2

6. What is a user ID (UID)? Why is UID=0 significant? What is a group ID (GID)?
   **Reference:** *Introduction to Linux* (http://www.tldp.org/guides.html)

7. What is system call 24? What is system call 47? What parameters (if any) do each take? Where are the return values (if any) stored?

8. Consider the following two programs:

   ```assembly
   .section .bss
   .lcomm buffer, 32000
   .section .text
   .globl _start
   _start:
       movl $1, %eax
       movl $0, %ebx
       int $0x80
   
   and
   
   .section .data
   buffer:
       .fill 32000
   .section .text
   .globl _start
   _start:
       movl $1, %eax
       movl $0, %ebx
       int $0x80
   ```

   Compile each, and compare the size of each program. Explain any differences you see, or explain why there should not be any difference.

9. Consider the program `indexed.s` presented in class. Modify only the text section so that only every second prime number is moved into `eax`. 

---