CS 35L Software Construction Laboratory (Lab6-A)
Mon, Oct 31, 2011, Ver 1.0

A brief introduction to Operating System

1. Process
   A process is an instance of a computer program that is being executed.
   It contains the program code and its current activity (memory).

   http://en.wikipedia.org/wiki/Process_(computing)
   http://www.comptechdoc.org/os/linux/usersguide/linux_ugprocesses.html

2. Kernel Mode and User Mode
   Ring 0 (Kernel)
   Ring 3 (Application)
   Kernel mode has more privileges:
     Unristricted mode. Full instruction availability. I/O
   operation, area of memory accessed are unlimitied.

   http://en.wikipedia.org/wiki/Ring_(computer_security)

3. System Call

   User processes ask the kernel to do things for them by invoking system calls.
     -- software interrupt, with a system call number
   When the user process invokes a system call, the processor enters kernel mode,
   the processor and the kernel cooperate to save the user process's state, and
   the kernel executes appropriate code in order to carry out the system call.
   When it's done, it resumes the user process.

   -- Unix/Linux System Call

    The system call number will go in %eax
    The arguments (up to five of them) will go in %edx, %ecx, %ebx, %edi,
  and %esi, respectively.
    The kernel passes the return value back in %eax

  #include <sys/syscall.h>
  syscall(SYS_ID, ...);
  Note: ... is the list of parameters
  %eax name   %ebx   %ecx   %edx   %esx %edi
  3   SYS_read   unsigned int char *   size_t -   -
  4   SYS_write unsigned int const char * size_t -   -

   http://bluemaster.iu.hio.no/edu/dark/lin-asm/syscalls.html

5. File Descriptor
   A file descriptor is an abstract indicator for accessing a file
   3 standard POSIX file descriptors
   Integer value   Name
   0   Standard Input (stdin)
   1   Standard Output (stdout)
   2   Standard Error (stderr)
   http://en.wikipedia.org/wiki/File_descriptor

```
   -- Input/Ouput redirection REVISITED:
   cat file1 > file2
   cat file1 1 > file2
   cat file1 1 > file2 2>&1

6. Wrapped System Call

   #include <unistd.h>
   ssize_t      read(int fd, void * buf, size_t sz);
   ssize_t      write(int fd, const void * buf, size_t sz);

   #include <sys/stat.h>
   int          fstat(int fd, struct stat *buf);
   http://pubs.opengroup.org/onlinepubs/009695399/functions/fstat.html

7. Tools for Lab 6

  Using strace to test the results.
    $ strace -o catb_trace ./catb < testfile

  Using time to measure the speed.
    $ time ./catb < testfile
```