LINUX ARM SYSCALLS

Basic syscalls in ARM Linux:

	Calling				Returns
Name	**Code in r7**	**r0**	**r1**	**r2**	**r0**
exit	1	int **exitCode**			
read	3	int **fileHandle**	char * **memory**	int **numBytes**	int **bytesRead**
write	4	int **fileHandle**	char * **memory**	int **numBytes**	
open	5	char * **fileName**	int **flags**	int **mode**	int **fileHandle**
close	6	int **fileHandle**			

File Handles:

Standard handles:

- 0 = console input (stdin)
- 1 = console output (stdout)
- 2 = console error stream (stderr)

For actual files, first do open syscall. It returns fileHandle number in r0. Store and use that number in read/write/open calls.

Flags Options:

Add together one or more of the following codes.
Note values are in Octal (base 8) make sure to include leading 0 when entering into assembly. Example: Write, Create,Truncate would be #01101

00	Read ONLY
01	Write ONLY
02	Read/Write
0100	Create file if does not exist
01000	Truncate existing file
02000	Append to existing file

Mode Options:

What permissions to give the new file.
0644 means you can read/write, everyone else just read (include leading 0 to indicate octal)
For more info see:
http://ryanstutorials.net/linuxtutorial/permissions.php