ARM Angel SWI Instruction Usage

All Angel SWI operations use the following instruction:

```
swi  0x123456
```

with register R0 containing the number of the requested operation. (R0 contains a *reason code*.) In most cases, register R1 contains the address of a block of parameters held in memory. Each parameter is one word (4 bytes) in size. On return from the SWI operation, R0 holds a result value.

A summary of the implemented SWI operations is provided in Table 1. The table is followed by more detailed descriptions

An example of using one of these SWI operations is shown below. It opens a text file for input and then reads 80 bytes from that file.

```
@   Example of using the Angel SWI operations

        ...                         @ omitted code

        ldr     R1, =OpenParams     @ parameters block for OPEN
        mov     R0, #0x01           @ code number for Open File
        swi     0x123456            @ open a text file for input
        cmp     R0, #0
        blt     OpenError           @ branch if there was an error
        ldr     R1, =ReadParams
        str     R0,[R1]             @ save the file handle into
                                    @   parameters block for READ
        mov     R0, #0x06           @ code number for Read File
        swi     0x123456            @ read from the text file
        cmp     R0, #0
        bne     ReadError           @ branch if there was an error

        ...                         @ omitted code

        .data
ReadParams:
        .word    0                  @ the file handle
        .word    InputBuffer        @ address of input buffer
        .word    80                 @ number of bytes to read
InputBuffer:
        .skip    80
OpenParams:
        .word   FileName
        .word   FileNameEnd-FileName   @ length of filename
        .word   0                   @ File mode = read
FileName:
        .ascii  "MyData.txt"        @ name without final NUL byte
FileNameEnd:
        .byte   0                   @ the NUL byte
```

Table 1: Summary of Angel SWI Operations

R0	R1[a]	Description	Operands in Memory (at address provided by R1)
0x01	M	Open a File	Filename address; filename length; file mode
0x02	M	Close a File	File handle
0x05	M	Write to File	File handle; buffer address; number of bytes to write
0x06	M	Read from File	File handle; buffer address; number of bytes to read
0x09	M	Is a TTY?	File handle
0x0A	M	File Seek	File handle; offset from file start
0x0C	M	File Length	File handle
0x0D	M	Temp File Name	Buffer address; unique integer; buffer length
0x0E	M	Remove File	Filename address; filename length
0x0F	M	Rename a File	Filename 1 address; length 1; Filename 2 address; length 2
0x10	–	Execution Time	
0x11	–	Absolute Time	
0x13	–	Get Error Num	
0x16	A	Get Heap Info	
0x18	Code	Exit Program	

a. M indicates the address of the block of operands in memory; A indicates the address of a four word block of memory to receive a result; Code indicates a termination code for the program.

Reason Code 0x01 (Open a File)

Parameters:

1	the address of a string in memory providing the filename
2	the number of characters in the filename
3	the file mode

The supported file mode values are

0x0	text input
0x1	byte input (i.e. raw mode)
0x4	text output (with file creation, or file truncation if file exists)
0x5	byte output (with file creation, or file truncation if file exists)
0x8	text output appending to a file
0x9	byte output appending to a file

The standard input, standard output and standard error output streams (stdin, stdout and stderr) are automatically opened by ARMSim. These streams do not have to be explicitly opened. They are accessed with file handles of 0, 1 and 2 respectively. If the program does need to explicitly open these streams, the filename `":tt"` should be used. (As explained below.)

As a special case, if the filename comprises the 3 characters `":tt"`, then the file handle refers to the console window. The first three calls to this SWI operation with the filename `":tt"` cause the console window to be associated with standard input (stdin), standard output (stdout) and standard error (stderr) in that order. The result returned in register R0 is 0, 1 or 2 respectively. The file mode parameter is ignored in this case (though standard C library functions use the mode value `0x3` which means byte input-output). The fourth, fifth ... calls to this SWI operation with the filename `":tt"` cause new console windows to be opened.

Result:

R0 contains a file handle (for use with other SWI calls) on success. If the operation fails, R0 holds -1 and an error code can be obtained with the 0x13 ErrorNo SWI call.

Reason Code 0x02 (Close a File)

Parameters:

1	a file handle

Result:

R0 holds 0 on success or an error code on failure.

Reason Code 0x05 (Write to a File)

Parameters:

1	a file handle
2	the address of a block of data in memory
3	the number of bytes to be written

Note: file handle values of 1 and 2 refer to the standard output and standard error output streams.

Result:

R0 holds the number of bytes NOT written. (I.e., if R0 is 0, the call was successful.) If R0 is non-zero, then an error code can be obtained with the 0x13 ErrNo SWI call.

Reason Code 0x06 (Read from a File)

Parameters:

1	a file handle
2	the address of a buffer in memory to receive the input
3	the number of bytes to be read

Note: a file handle value of 0 refers to the standard input stream.

Result:

R0 holds the number of bytes NOT read. (I.e., if R0 is 0, the call was successful.) If R0 is non-zero, then an error code can be obtained with the 0x13 ErrNo SWI call.

Reason Code 0x09 (Is It a TTY Test)

Parameters:

1	a file handle

Result:

R0 holds 1 if the file handle refers to the ARMSim console, and 0 otherwise. If the call fails, then R0 holds -1 and an error code can be obtained with the 0x13 ErrNo SWI call.

Reason Code 0x0A (File Seek)

Parameters:

1	a file handle
2	the absolute position in bytes (i.e. offset from the file start) to seek to

Result:

R0 holds 0 if the seek is successful. If the call fails, then R0 holds -1 and an error code can be obtained with the 0x13 ErrNo SWI call.

Reason Code 0x0C (File Length)

Note: this operation is currently implemented only for input files.

Parameters:

1	a file handle

Result:

R0 holds the length of the file in bytes (or 0 if the file handle refers to the ARMSim console). If the call fails, then R0 holds -1 and an error code can be obtained with the 0x13 ErrNo SWI call.

Reason Code 0x0D (Create a Name for a Temporary File)

Parameters:

1	the address of a buffer in memory to receive the file name
2	an integer in the range 0 to 255 to distinguish one temporary file name from another
3	the size of the buffer in bytes

Result:

On success, the buffer holds a null-terminated string which can be used as a temporary file name. The leading part of the string is the same for all file names generated with this SWI operation. The file name ends with the digits of the second parameter. Therefore the second parameter is important if more than one temporary file is used in the program.

R0 holds 0 on success. If the call fails, then R0 holds -1 and an error code can be obtained with the 0x13 ErrNo SWI call.

Reason Code 0x0E (Remove a File)

Parameters:

1	the address of a string in memory providing the filename
2	the number of characters in the filename

Result:

R0 holds 0 if the file was successfully deleted. If the call fails, then R0 holds -1 and an error code can be obtained with the 0x13 ErrNo SWI call.

Reason Code 0x0F (Rename a File)

Parameters:

1	the address of a string in memory providing filename #1
2	the number of characters in filename #1
3	the address of a string in memory providing filename #2
4	the number of characters in filename #2

Result:

An existing file specified by the first filename is renamed to have the second filename. If the operation is successful, R0 is 0. If the call fails, then R0 holds -1 and an error code can be obtained with the 0x13 ErrNo SWI call.

Reason Code 0x10 (Get User-Mode Execution Time)

Parameters: None

Result:

R0 holds the user-mode execution time as a multiple of 0.01 second units.

Reason Code 0x11 (Get Absolute Time)

Parameters: None

Result:

R0 holds the number of seconds since the Unix epoch, which is at 0:00 on 1 January 1970.

Reason Code 0x13 (Get Error Number)

Parameters: None

Result:

R0 holds an error number. The error numbers are listed in Table 2. Note that many of these error numbers cannot occur. All those that can occur with the current implementation of the Angel SWI operations on ARMSim are indicated in the righthand column of the table. (The unused error codes could be supported in a future version of ARMSim.)

Reason Code 0x16 (Get Heap Information)

Parameters:

1	the address of a four word array in memory

Result:

The four word array is filled with the following four numbers: the address of the start of the area of memory reserved for the heap, the address of the end of that heap area, the address of the area of memory reserved for the stack, the address of the end of that stack area.

R0 holds 0 if the operation is successful. If the call fails, then R0 holds -1 and an error code can be obtained with the 0x13 ErrNo SWI call.

Reason Code 0x18 (Exit the Program)

Parameters:

Register R1 holds a program termination code.

If the code in R1 is `0` or `0x20026`, then the program terminates normally. Otherwise an error message containing the termination code is displayed and the program still terminates.

Result: None

There is no result returned from the call because the program is halted.

Table 2: Codes Obtained by the Get Error Number Operation

Code and Unix Name	Description	Can Occur?
NONE = 0	No error to report	Y
ENOENT = 2	No such file or directory	
EIO = 5	I/O error	Y
EBADF = 9	Bad file number	Y
ENOMEM = 12	Not enough memory to complete operation	
EACCES = 13	Permission denied	
EFAULT = 14	Bad address	Y
EEXIST = 17	File exists	
ENOTDIR = 20	Not a directory	
EISDIR = 21	Is a directory	
EINVAL = 22	Invalid argument	Y
EMFILE = 24	Too many open files	
ENOTTY = 25	Not a TTY console	
EFBIG = 27	File too large	
ESPIPE = 29	Illegal seek	
EROFS = 30	Read only file system	
ENOTSUP = 134	Not supported	Y