

Argante2

Syscall Reference

James Kehl
ecks@optusnet.com.au

May 26, 2002

Chapter 1

Syscall Reference

Most syscalls will take their first argument in r0, second argument in r1, and so on. A syscall marked SYS2 will use an extra syscall2 argument as its first input, and take any extra arguments from r1 on.

The syntax used to describe inputs and outputs is borrowed from LAC. A "string" is two arguments: a buffer address, and that buffer's size.

1.1 ALIB

Author: James Kehl <ecks@optusnet.com.au>
Editor: James Kehl <ecks@optusnet.com.au>
Last Updated: Sun Jan 13 2002

Provides syscalls for run-time linking.

Known Issues:

ALIB_LOOKUP doesn't say whether symbol is code or data.

1.1.1 ALIB_OPEN

Inputs: ["library-filename"]

Outputs: [unsigned library_handle]

SideFx: Loads code, data & symbols from specified image.

Other: Can throw ERR_ALIB_FAIL

1.1.2 ALIB_LOOKUP (SYS2)

Inputs: [unsigned library_handle, "symbol name"]

(A zero library-handle means any)

Outputs: [unsigned symbol_address]

SideFx: N/A

Other: Throws ERR_ALIB_FAIL on a bad lib handle

Throws ERR_ALIB_NOSYM if symbol not found or undefined

1.1.3 ALIB_CLOSE (SYS2)

Inputs: [unsigned library_handle]

Outputs: []

SideFx: Unloads pages, data & symbols belonging to specified lib

Other: Throws ERR_ALIB_FAIL on a bad lib handle

1.2 CFD

Author: James Kehl <ecks@optusnet.com.au>
Editor: James Kehl <ecks@optusnet.com.au>
Last Updated: Sat 25 May 2002

The Common File Descriptor operations module allows generic access to virtual file descriptors.

Known Issues:

Not all calls are implemented yet.

1.2.1 CFD_CLOSE (SYS2)

Inputs: [unsigned vfd_handle]

Outputs: []

SideFx: Closes and invalidates vfd_handle.

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

1.2.2 CFD_READ (SYS2)

Inputs: [unsigned vfd_handle, "writable buffer..."]

Outputs: [ignore, unsigned @null_after_data, unsigned space_left_in_buffer]

SideFx: Reads up to r2 bytes from vfd_handle

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

1.2.3 CFD_WRITE (SYS2)

Inputs: [unsigned vfd_handle, "buffer..."]

Outputs: [ignore, unsigned @null_after_data, unsigned bytes_unwritten]

SideFx: Writes up to r2 bytes from vfd_handle

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

1.2.4 CFD_WRITE_NT (SYS2)

Inputs: [unsigned vfd_handle, "bufferbufferbuffer\000"]

Outputs: [ignore, unsigned @null_after_data, unsigned bytes_unwritten]

SideFx: Writes r2 bytes or up to the first null, whatever comes first.

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

1.2.5 CFD_WRITE_CHAR (SYS2)

Inputs: [unsigned vfd_handle, unsigned char]

Outputs: []

SideFx: Writes low byte of char to VFD.

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

1.2.6 CFD_WRITE_FLOAT (SYS2)

Inputs: [unsigned vfd_handle, float a, unsigned min_digits,
unsigned max_digits]

Outputs: []

SideFx: Writes a to VFD in "%g" format.

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

Throws ERR_ARG_TOOLONG if string would be more than 32 characters, or
max_digits or min_digits are larger than 8.

1.2.7 CFD_WRITE_INT (SYS2)

Inputs: [unsigned vfd_handle, signed a]

Outputs: []

SideFx: Writes a to VFD as a signed int.

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

Throws ERR_ARG_TOOLONG if string would be more than 16 digits.

1.2.8 CFD_WRITE_UINT (SYS2)

Inputs: [unsigned vfd_handle, unsigned a]

Outputs: []

SideFx: Writes a to VFD as a unsigned int.

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

Throws ERR_ARG_TOOLONG if string would be more than 16 digits.

1.2.9 CFD_WRITE_HEX (SYS2)

Inputs: [unsigned vfd_handle, unsigned a]

Outputs: []

SideFx: Writes a to VFD in hexadecimal format.

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

Throws ERR_ARG_TOOLONG if string would be more than 16 digits.

1.2.10 CFD_WRITE_OCT (SYS2)

Inputs: [unsigned vfd_handle, unsigned a]

Outputs: []

SideFx: Writes a to VFD in octal format.

Other: Throws ERR_BAD_FD if file descriptor does not support this operation.

Throws ERR_ARG_TOOLONG if string would be more than 16 digits.

1.3 DISPLAY

Author: Unknown
Editor: James Kehl <ecks@optusnet.com.au>
Last Updated: Fri Jul 13 2001

Simple console output module for debugging.

Known Issues:

This will be phased out when agents are introduced.

Does not flush output until newline printed.

1.3.1 IO_PUTSTRING

Inputs: ["string"]

Outputs: []

SideFx: Displays string on kernel console.

Other: N/A

1.3.2 IO_PUTINT (SYS2)

Inputs: [signed a]

Outputs: []

SideFx: Displays signed integer on kernel console.

Other: N/A

1.3.3 IO_PUTCHAR (SYS2)

Inputs: [unsigned a]

Outputs: []

SideFx: Displays raw 8 low bits of a on console.

Other: "syscall2 \$IO_PUTCHAR, 10" will print newlines and flush output.

1.3.4 IO_PUTFLOAT (SYS2)

Inputs: [float a]

Outputs: []

SideFx: Displays float in "%g" format.

Other: N/A

1.3.5 IO_PUTHEX (SYS2)

Inputs: [unsigned a]

Outputs: []

SideFx: Displays hex number.

Other: N/A

1.4 FS

Author: James Kehl <ecks@optusnet.com.au>
Editor: James Kehl <ecks@optusnet.com.au>
Last Updated: Fri Jul 13 2001

Interaction with filesystem and IO to real files.

ERR_ARG_TOOLONG is thrown when a filename is larger than PATH_MAX.
 ERR_BAD_FD occurs when you try and read or write something you didn't
 open for that (incl. directories).

Known Issues:

Does not support nonblocking IO.

Open flags are possibly more confusing than real system's.

Truncation doesn't happen yet. ****FIXME****

Partial write/read only returns errors, doesn't throw them.

ERR_GENERIC happens a little too often. ****FIXME****

1.4.1 FS_OPEN (SYS2)

Inputs: [unsigned open_mode, "filename"]

Outputs: [unsigned vfd_handle]

SideFx: Creates file if not existing.

Other: Access modes:

0 - APPEND mode

1 - READ flag

2 - WRITE flag

4 - FSEEK flag

Opening a file for plain WRITE will truncate the file. (soon.)

1.4.2 FS_OPEN_EXISTING (SYS2)

Inputs: [unsigned access_mode, "filename"]

Outputs: [unsigned vfd_handle]

SideFx: Fails with ERR_FILE_NOT_EXIST if not existant.

Other: Access modes as above.

1.4.3 FS_OPEN_CREATE (SYS2)

Inputs: [unsigned access_mode, "filename"]

Outputs: [unsigned vfd_handle]

SideFx: Creates file, fails with ERR_FILE_EXIST if file already exists.

Other: Access modes as above.

1.4.4 FS_READ (SYS2)

Inputs: [unsigned vfd_handle, "writable buffer..."]

Outputs: [ignore, unsigned @null_after_data, unsigned space_left_in_buffer]

SideFx: Reads at most r2 bytes into buffer.

Other: N/A.

1.4.5 FS_WRITE (SYS2)

Inputs: [unsigned vfd_handle, "buffer to write..."]

Outputs: [ignore, unsigned @null_after_data, unsigned bytes_unwritten]

SideFx: Writes r2 bytes to given file.

Other: N/A.

1.4.6 FS_FLUSH (SYS2)

Inputs: [unsigned vfd_handle]

Outputs: []

SideFx: All bytes written are immediately stored, and any bytes written by another process are read (if applicable).

Other: N/A.

1.4.7 FS_SEEK (SYS2)

Inputs: [unsigned vfd_handle, unsigned new_offset]

Outputs: []

SideFx: Current file position (for reading/writing) or directory handle will be set to new_offset. Sparse files may be created.

Other: Only does SEEK_SET.

1.4.8 FS_TELL (SYS2)

Inputs: [unsigned vfd_handle]

Outputs: [unsigned file_pos]

SideFx: None.

Other: May be used on directories, but value is not guaranteed to be consistent.

1.4.9 FS_WD_GET

Inputs: ["writable buffer for working directory"]

Outputs: [ignore, unsigned path_length]

SideFx: Puts working directory path in buffer.

Other: N/A.

1.4.10 FS_WD_SET

Inputs: ["new working directory"]

Outputs: []

SideFx: Changes working directory

Other: N/A.

1.4.11 FS_OPEN_DIR

Inputs: [ignore, "directory name"]

Outputs: [unsigned vfd_handle]

SideFx: None past the obvious.

Other: N/A.

1.4.12 FS_READ_DIR (SYS2)

Inputs: [unsigned vfd_handle, "writable buffer..."]
Outputs: [ignore, ignore, unsigned filename_length]
SideFx: Puts the next filename in buf.
Other: N/A.

1.4.13 FS_MAKE_DIR

Inputs: ["filename"]
Outputs: []
SideFx: You reckon?
Other: N/A.

1.4.14 FS_STAT

Inputs: ["filename"]
Outputs: [unsigned file_type, unsigned file_size, unsigned mod_time]
SideFx: None.
Other: Filetype is -
0 - error
1 - regular file
2 - directory
3+- something else. (pipe?)

1.4.15 FS_RENAME

Inputs: ["source filename", "destination filename"]
Outputs: []
SideFx: One hopes so.
Other: Unlike C/Unix function of same name, fails if destination already exists.

1.4.16 FS_DELETE

Inputs: ["filename"]
Outputs: None.
SideFx: File/(empty) directory liquidation.
Other: None.

1.5 LOCALLIB

Author: Michal Zalewski <lcamtuf@coredump.cx>
Editor: James Kehl <ecks@optusnet.com.au>
Last Updated: Fri Jul 13 2001

Returns useful information on local system, and a few related utility functions.

Known Issues:

ERR.GENERIC happens a bit too often. ****FIXME****

RS_STAT is Linux only and somewhat dusty anyway.

VS_STAT is Argante1 only.

1.5.1 LOCAL_GETTIME

Inputs: []

Outputs: [unsigned seconds_since_epoch, unsigned μ secs]

SideFx: None.

Other: N/A.

1.5.2 LOCAL_TIMETOSTR (SYS2)

Inputs: [unsigned seconds_since_epoch, "writable buffer"]

Outputs: [ignore, ignore, unsigned bytes_stored]

SideFx: 'ctime' into buffer.

Other: N/A.

1.5.3 LOCAL_GETHOSTNAME

Inputs: ["writable buffer"]

Outputs: [ignore, unsigned bytes_stored]

SideFx: None.

Other: Any hostname longer than 64 characters is 'bogus' in more ways than one.

1.5.4 LOCAL_GETRANDOM

Inputs: []

Outputs: [unsigned random_integer]

SideFx: None.

Other: N/A.

1.5.5 LOCAL_RS_STAT

Inputs: []

Outputs:

[unsigned uptime, unsigned 5sec_load_average, unsigned total_ram_in_K, unsigned free_ram, unsigned total_swap, unsigned free_swap, unsigned processes]

SideFx: None.

Other: N/A.

1.6 STRFD

Author: James Kehl <ecks@optusnet.com.au>
Editor: James Kehl <ecks@optusnet.com.au>
Last Updated: Sat 5 May 2002

The StrFD module allows manipulation of a block of memory using a read / write / seek paradigm and the Common File Descriptor operations set, and is probably the easiest way to access memory on a byte-by-byte basis.

Known Issues:

ERR_STRFD_BOUNDS occurs when the offset exceeds the allocated size.

ERR_STRFD_SEARCHFAIL occurs when one of the search functions fail. The offset is also invalid when this occurs and should be reset.

1.6.1 STRFD_OPEN

Inputs: ["existing buffer"]

Outputs: [unsigned vfd_handle]

SideFx: Use this block as a static, unresizable StrFD.

Other: N/A.

1.6.2 STRFD_CREATE

Inputs: [unsigned initial_size]

Outputs: [unsigned vfd_handle]

SideFx: Create a new memory block attached to a dynamically-grown StrFD.

Other: N/A.

1.6.3 STRFD_CLOSE (SYS2)

Inputs: [unsigned vfd_handle]

Outputs: []

SideFx: Closes the given StrFD.

Other: Does NOT free dynamically-allocated memory, so be sure to get the address first.

1.6.4 STRFD_GET_OFFSET (SYS2)

Inputs: [unsigned vfd_handle]

Outputs: [unsigned offset]

SideFx: None.

Other: This returns the byte offset within the memory block for which the next read or write will start from.

1.6.5 STRFD_SET_OFFSET (SYS2)

Inputs: [unsigned vfd_handle, unsigned offset]

Outputs: []

SideFx: None.

Other: This sets the byte offset for which the next read or write will start from.

1.6.6 STRFD_GET_ADDR (SYS2)

Inputs: [unsigned vfd_handle]

Outputs: [unsigned addr]

SideFx: None.

Other: This is most useful for accessing dynamically allocated strings outside the StrFD module.

1.6.7 STRFD_GET_SIZE (SYS2)

Inputs: [unsigned vfd_handle]

Outputs: [unsigned size]

SideFx: None.

Other: This is the flip side of STRFD_GET_ADDR.

1.6.8 STRFD_READ (SYS2)

Inputs: [unsigned strfd_handle, "writable buffer..."]

Outputs: [ignore, unsigned @null_after_data, unsigned space_left_in_buffer]

SideFx: Reads up to r2 bytes from strfd_handle and shifts the offset.

Other: N/A.

1.6.9 STRFD_WRITE (SYS2)

Inputs: [unsigned strfd_handle, "buffer..."]

Outputs: [ignore, unsigned @null_after_data, unsigned bytes_unwritten]

SideFx: Writes up to r2 bytes from strfd_handle and shifts the offset

Other: N/A.

1.6.10 STRFD_GETCHAR (SYS2)

Inputs: [unsigned strfd_handle]

Outputs: [unsigned a_byte]

SideFx: Increments the offset.

Other: N/A.

1.6.11 STRFD_SETCHAR (SYS2)

Inputs: [unsigned strfd_handle, unsigned a_byte]

Outputs: []

SideFx: Writes the single byte and increments the offset.

Other: N/A.

1.6.12 STRFD_STRCHR (SYS2)

Inputs: [unsigned strfd_handle, signed shift_val, unsigned needle]

Outputs: []

SideFx: Shifts the offset until STRFD_GETCHAR would return needle.

Other: Throws ERR_STRFD_SEARCHFAIL if character not found.

1.6.13 STRFD_STRSTR (SYS2)

Inputs: [unsigned strfd_handle, signed shift_val, "needle"]

Outputs: []

SideFx: Shifts the offset until STRFD_READ would return needle.

Other: Throws ERR_STRFD_SEARCHFAIL if string not found.

1.6.14 STRFD_STRCMP (SYS2)

Inputs: [unsigned strfd_handle, "needle"]

Outputs: [signed difference]

SideFx: None - does not change offset.

Other: Returns any differences between needle and the next r2 bytes of the StrFD, so could also be called strcmp or memcmp.

1.6.15 STRFD_SPN (SYS2)

Inputs: [unsigned strfd_handle, "accept"]

Outputs: []

SideFx: Increases offset until STRFD_GETCHAR would return a character not in "accept".

Other: Not Implemented. ****FIXME****

1.6.16 STRFD_CSPN (SYS2)

Inputs: [unsigned strfd_handle, "reject"]

Outputs: []

SideFx: Increases offset until STRFD_GETCHAR would return a character in "reject".

Other: Not Implemented. ****FIXME****