1616: Quick Reference Guide

Version 4.059
August 1993

Applix 1616 microcomputer project
Applix pty limited
Lot 1, Kent St., Yerrinbool, NSW 2575

Applix 1616 Computer System

Do you want a standard MS-DOS or Macintosh computer system full of custom ASIC chips and undocumented features? Do you want to deal with sales people who know little more than the price of the computer? Do you want to buy expensive programs, and then find the dealer knows nothing that isn't obvious from the manual. If so, don't bother to read this.

Or do you like to understand every single chip in your system, have every function accessible and changeable, and have interface facilities readily available? If you have a difficult problem, would you like to talk to the person who designed the computer? Would you like to read the source code for the programs you are using? Would you like to build your own custom computer, or have one assembled to suit your needs?

The Applix 1616 is an Australian designed and built computer system for engineers, programmers, advanced students, and DIY enthusiasts. Particularly suited to custom interfaces, industrial control, and education, it makes a fine general purpose personal computer system. Provides a powerful EPROM resident multiuser, multitasking operating system, not unlike Unix, with lots of interface facilities. Accepts industry standard peripherals, no hard to get add-ons. Built with common TTL and LSI electronic components, no special parts (except for two 16R8 PAL chips). All facilities open and accessible from C, assembler, Forth, BASIC, or straight from the keyboard.

The 7.5 MHz (field upgradable to 15 MHz) Motorola 68000 or 68010 based motherboard is available as a bare board, in kit form, or fully built, to suit your budget and intentions. Provision on board for 512k RAM, dual Z8530 SCC serial ports (as in Macintosh), latched Centronics compatible printer port, Apple compatible joystick port, 3000 baud cassette interface, stereo sound outputs from 2 watt amplifiers, and accepts a standard IBM keyboard.

Plenty of uncommitted I/O for interfacing. 8 digital I/O lines from 6522 VIA, two 8 bit analog output lines, six 8 bit analog input lines. Four 80 pin expansion sockets (one used by disk controller) give access to all 68000 lines, including interrupts, plus power. Comes with all schematics, full circuit description, construction manual, and built in tests for fault finding. Tests (or complete construction) do not require any peripherals (a logic probe, keyboard and video monitor, or serial terminal, do help however).

Bit mapped 6545 based programmable video circuit on board runs IBM CGA monitors, or composite black and white. Software selectable 320 by 200 pixels in 16 colours, 640 by 200 in 4 colours from palette of 16. One additional TTL chip provides two additional modes, which require dual scan Hercules, EGA or multisync monitor. 320 by 350 in 16 colours, 640 by 350 in four colours from a pallette of 16. Second PAL chip provides additional 960 by 512 monochrome display, with software support for multiple windows (requires multisync or 30 KHz monitor). Or, if you like, run the entire system from a serial terminal without video.

Multiuser (3 users), multitasking (64 tasks) 1616/OS operating system, with over sixty commands, plus over 200 system calls, in the onboard 128k of EPROM. EPROM contains a WordStar compatible full screen editor, haa complete 68000 macro assembler with include file facilities and conditional assembly, plus ASCII tables, an expression evaluator, timing of commands, clock facilities, a small machine code monitor, and many utilities. Terminal emulator provides Televideo 950 compatible terminal facilities via both serial ports. File commands handle onboard ram disk, optional floppy drives, and SCSI hard disks, with a hierarchical file system. Cold boot from floppy, hard disk, EPROM card, or without any drives at all. Warm boot from any drive, including ram disk. User Tutorial, User Reference manual, Programmers manual, and Technical Reference manuals included. Many other manuals available.

Plug in intelligent disk drive controller card has 8 MHz Z80, sockets for 64k of RAM (8k standard), 32k EPROM, WD1770 floppy disk controller chip, socket for NCR 5380 SCSI hard disk controller chip, socket for Z8530 SCC dual serial ports. Run any 40 or 80 track 3.5 inch or 5.25 inch double density floppy drives (two 800k 3.5 inch double sided 80 track drives are recommended). Can run any hard disk with inbuilt SCSI, or use Adaptec, Xebec or similar SCSI to ST506 converter to run IBM style ST506 hard disk. Disk controller can optionally run ZCPR ZRDOS CP/M emulation, using Microbee format disks. WordStar, etc available. Free utility software can read, write and format MS-DOS disks, read and write many CP/M formats.

Using reset control

First try Alt C to interrupt the program, and if that fails, Alt Ctrl C.

Level 2 reset is Alt Ctrl R, or press hardware Reset switch. Ram disk contents are retained.

Two resets (by either method) within three seconds of each other do a Level 1 reset, which does extra initialisation. Or type `syscall 1` from keyboard.

The 'level 0' reset should occur only at power-on time. A level 0 reset reinitialises everything, including the RAM disk. Or type `syscall .101` from keyboard.

Control and Alt key

Ctrl key plus another key generates the ASCII code of that key, minus 64, providing a Control code between 0 and 31.

Alt key plus another key generates the ASCII code of the key, plus 128 (for programmers, bit 7 is set by the Alt key).

All 256 characters of the extended ASCII character set are obtainable by holding down the Alt key and typing the character's ASCII code (in decimal) on the numeric keypad section of the keyboard.

The Alt key

Special functions use the Alt key:

Alt T	Toggles the cassette relay.
Alt Ctrl R	Level 2 reset.
Alt S	Stops and starts character output on video output. Cursor changes to an underline to indicate that output is suspended.
Alt C	Abort from a program or a function.
Alt Ctrl C	Abort any program in foreground.
Alt Del	Toggle EOF character between $04 and $100 (default is $100).

Function key macros

The ten function keys F1 to F10 may be used to produce up to 63 characters of input from a single keystroke, providing macros for any program.

Hold down the Alt and Ctrl and press the function key. From this point all characters typed are invisibly captured into the definition for that function key. Terminate definition by pressing the function key which is being defined.

Last line recall

The up-arrow ↑ and down-arrow ↓ keys (or Ctrl E and Ctrl X) scroll up and down through the last 10 lines which have been entered in the line editor. Once you have found the line you wish, you can change it with the powerful line editor.

Typing the first few characters of a past line, then using the Esc key will complete the line. Press Esc again for the next line that matches.

Cursor positioning commands:

Ctrl D or →	Go forward one character
Ctrl F	Go forward one word
Ctrl S or ←	Go backward one character
Ctrl A	Go backward one word
Ctrl B or End	If at start of line, go to end; otherwise go to start

Text deletion commands:

Ctrl G or Del	Delete the character under the cursor
Ctrl T	Delete from the cursor to the start of the next word
Ctrl Y	Delete from the cursor to the end of the line
Bs or Ctrl H	Delete the character before the cursor
Ctrl V	Delete from the start of the line up to the cursor

Miscellaneous commands:

Ctrl W	Recall previously typed lines, insert at cursor
Ctrl U	Undo the last deletion, insertion or command (some versions only)
Ctrl P	Escape a control character - after typing a Ctrl P you may enter any single control character into the line.
Ctrl M or Enter	The Enter key is pressed when you are satisfied with the line. The cursor may be at any point on the line when Enter is pressed. On some keyboards, the Enter key is marked Return.
Ctrl E or ↑	Scroll backwards through previous lines
Ctrl X or ↓	Scroll forwards through previous lines

4 Connections

Looking at the rear, and viewing from left to right, the connectors are:

- Keyboard, IBM XT standard (5 pin DIN socket)
- Loudspeakers, 2 watt stereo output (5 pin DIN socket)
- Cassette, with motor control (5 pin DIN socket)
- Reset button (at rear of board)
- Joystick Port, for standard Apple joystick (9 pin D socket)
- Serial Port B, custom layout can provide +12v, -12v (9 pin D plug)
- Serial Port A, (9 pin D plug)
- Video Connector, identical to IBM CGA (9 pin D socket)
- User Control Port (34 way plug)
- Centronics Parallel Printer Port, use ribbon cable to printer (26 way plug)
- Power Switch
- Power Line Connector (3 pins in rectangular socket)

- Internal connection pins for unamplified sound outputs.
- Internal jumper access to Z8530 SCC raw inputs and outputs, for use with Appletalk or other (non RS232C) serial protocols.
- User Control Port has 8 digital I/O lines to 6522 VIA chip, two analog outputs, 6 analog inputs, the Centronics Ack line (which generates an interrupt), plus power and ground (+5, -5, +12, -12 volts). Connect microphone for sound digitising to input pin 27, ground pin 25, or stereo microphone to input pins 27 and 28, ground pin 25).

Special characters

; comment on command lines, rest of line is ignored.

" ignore special characters enclosed in quotes.
' 'A 'B etc., enter ASCII code.

! separate multiple commands, up to 511 character allowed in line.

. Decimal numbers are preceded by **.**
% Binary numbers are preceded by **%**
- Minus may precede decimal number only (e.g. .-5)
Hexadecimal numbers have no leading character.

***** Wildcard for any group of characters, works in all commands.
? Wildcard for any single character.
[] Group characters.
~ Negate a group.

| Pipe output of command to input of next command.
& Run command asynchronously, in background (multitasking).

Input / output redirection

commandname arguments <inputsource > outputdestination **}** errordestination

The optional input / output redirections '<', '>' and '**}**' in the command line format above will get input from, and send output and error messages to, the nominated character devices or files for the duration of the command. Doubled redirections >> and **}}** means that new output is appended to the previous output file, rather than overwriting it.

The '<', '>', '>>', '**}**' and '**}}**' constructs must be the last part of the command line; all characters after these are ignored. Using wildcards in I/O redirection filenames will not work; the whole filename must be entered.

Character devices

A character device is identified by a name followed by a colon.

CON: Console, the video display when used as output, and the keyboard when used as input.

SA: Serial channel A for input and output.

SB: Serial channel B for input and output.

CENT: Centronics parallel printer output port. Input may not be obtained from this device.

NULL: Discards characters which are sent to it. Input may not be obtained from this device.

Some examples of commands which employ I/O redirection follow:

```
dir >myfile
dir >>myfile
edit myfile <edcommandfile
```

This command uses the full-screen editor upon the file `myfile`. The file `edcommandfile` would contain a sequence of characters which are presented to the editor as if you had typed them in.

```
SSASM asmfile.s -l >CENT: }errorlog
```

parallel (centronics) printer port. Error messages are recorded in the file `errorlog` in the current directory.

Block devices, disk drives

Block devices are **/RD**, ram disk. **/F0** and **/F1**, floppy disk drives, **/H0**, **/H1**, etc hard drives, **/S0**, etc fast SCSI. Don't forget the **/**.

RAM disk

Vary the amount of memory which 1616/OS allocates for the RAM disk using switches 0 and 1 of the quad switch on the 1616's PCB as below (exact ram may vary from version to version). MRDRIVERS software can override this on boot up, if required.

RAM disk	Switch 1	Switch 0	RAM disk size
0	off	off	24K
1	off	on	104K
2	on	off	200K
3	on	on	304K

Search path

- **current** directory for `.xrel`, then for `.shell`, then `.exec`.
- All directories in `xpath`, as above (see `xpath` command).
  The order of these can be swapped, see `option .19 2`.

Shell programs

; Comments are preceded by a semicolon.

$1 first argument.
$2 second argument, etc.
$0 returns the name of the `shell` file itself.
$* all arguments except the name of the `shell` file (arguments 1 and on).

`trap` enables error trapping.
`notrap` disable error trapping mode.
`trap2` non-zero error trapping.
`+` enable command echoing.
`-` disable command echoing.
command args <<end_marker redirect this from standard input end_marker

All commands are entered by typing their name. Options and filesnames are separated from the command by space(s).

Command format

commandname arguments <inputsource > outputdestination **}** errordestination

FILE RELATED

Manipulate files on disk devices. Also apply to character devices such as CON: and CENT:.
The COPY, MOVE, TYPE, CAT and CIO commands overlap in their functions and there are a number of ways of doing any one thing. Terminate with an end-of-file character, if set, usually Ctrl d.

Copying, joining, moving files

CAT [pathname1] [pathname2] [device1:] [device2:] ...

Copying files and directories

COPY source1 [source2] ... destination

Moving files and directories

MOVE source destination (original file is removed)

Displaying files

TYPE pathname1 [pathname2] ... Accepts wildcards, multiple files.
TYPE device: [pathname1] ...

Deleting files and empty directories

DELETE pathname1 [pathname2] [pathname3] ...

Renaming files and directories

RENAME pathname filename

Change date of file or directory

TOUCH pathname1 [pathname2] ...

Changing file attributes

FILEMODE 0 mask file1 [file2] ... Clear attribute bits
FILEMODE 1 mask file1 [file2] ... Set attribute bits

0	$0001	Backup bit: the file is backed up
1	$0002	Directory bit: the directory entry refers to a directory
2	$0004	Locked bit: the file is locked (read - only)

Bits 3, 4 and 5 are Read, Write and eXecute bits, displayed as RWX in DIR.
Bits 6, 7 and 8 are symbolic links, small file speedup valid, and hidden files.
Bit 9 is boring bit, files not to be backed up.

DIRECTORY RELATED

Directory listings

DIR [pathname1] [pathname2] ... Wildcard gives next level directories also.
DIRS [pathname1] [pathname2] ... Gives short, names only listing.

```
DIR /F0/*
```

Changing directory

CD	Display name of current directory
CD path	Change to a new directory
`CD /f0`	Makes the root directory of floppy 0 current
`CD ..`	Moves up a directory level
`CD mydir`	Moves down a level
`CD ../mydir`	Moves across a level

Creating a directory

MKDIR path	Make a directory

Execution search path

XPATH	Display path setting
XPATH -	Clear all paths
XPATH path1 [path2] ...	Set paths
XPATH + path1 [path2] ...	Add paths

Block device information

VOLUMES	Lists the volume name of every disk on line.

Substituting pathnames

ASSIGN	display all current assignments
ASSIGN -	delete all current assignments
ASSIGN /path1 /path2	/path2 is substituted whenever /path1 appears

Set environment string

SET	display all current settings
SET -	delete all current assignments
SET **-e** "name1=setting1"	set name1 to setting1, eg set -e del=delete
SET **-a** name1=setting1	for use anywhere in command, eg set -a work=$home/mydir

Saving files on tape

TSAVE pathname1 [pathname2] [pathname3] ...
First rewind the tape, put the player into record mode and allow the tape to move forwards until the leader is no longer over the tape head. Use Alt T command to toggle the cassette relay.

Archiving files on tape

TARCHIVE pathname1 [pathname2] [pathname3] ...

Loading files from tape

TLOAD [pathname1]

Loading multiple tape files

ITLOAD Mnemonic: Indefinite Tape LOAD

Verifying tape files

TVERIFY

MEMORY MANIPULATION

A safe memory area for experimenting with these commands is the $8000-$10000 memory range. The Alt C terminates memory examination. Commands start at address ‘a1’, end at address ‘a2’, use last address accessed as default for start. Some provide a prompted mode if no parameters are given.

Examining memory

MDB [a1] [a2]
MDW [a1] [a2]
MDL [a1] [a2] Mnemonic: Memory Dump (Byte, Word or Long)

Continuous memory examination

MRDB a1
MRDW a1
MRDL a1 Mnemonic: Memory Repetitively Display (Byte, Word, Long)

Memory alteration

MWB a1 [n1] [n2] [n3] ...
MWW a1 [n1] [n2] [n3] ...
MWL a1 [n1] [n2] [n3] ... Mnemonic: Memory Write (Byte, Word, Long)

Putting ASCII strings in memory

MWA a1 [string]
MWAZ a1 [string] Mnemonic: Memory Write Ascii (Zero)

Memory Filling

MFB a1 a2 n1
MFW a1 a2 n1
MFL a1 a2 n1
MFA a1 a2 string Mnemonic: Memory Fill (Byte, Word, Long, Ascii)

Memory comparing

MCMP a1 a2 a3 Mnemonic: Memory CoMPare
The memory blocks between addresses ‘a1’ through to ‘a2’ is compared to the blocks starting at address ‘a3’.

Memory searching

MSEARCH a1 a2 n1 [n2] [n3] ...

Memory moving

MMOVE a1 a2 a3

Saving memory in a file

MSAVE a1 a2 pathname1

Loading memory from a file

MLOAD pathname1 [a1]

COMMAND LINE REDIRECTION

Moving characters about

CIO [n1] Mnemonic: Copy Input to Output
Reads characters from standard input and writes them to standard output, optionally terminating on the character whose ASCII code is ‘n1’. Terminate on Alt Ctrl C

```
CIO 1a <sa: >myfile
```
Read from serial channel A and write onto new file ‘myfile’. Terminate on a control-Z (ASCII code $1a).

```
CIO >>myfile <sb:
```
Read characters from serial channel B and append then to ‘myfile’.

```
CIO <sa: >sb:
CIO <myfile >cent:
```

SHELL FILES

Echo command line arguments

ECHO [-n] [arg1] [arg2] ... (-n means no new-line)

Pausing

PAUSE n1 (cease processing for n1/50 seconds)

SYSTEM

Setting the time and date

SETDATE year month day hour minute second. Don't forget decimal point.

Displaying the current time/date

DATE

Executing machine code

GO a1 Perform an MC68000 'JSR' to 'a1'.

Manually performing system calls

SYSCALL callno n1 n2 n3 ...
This command causes 1616/OS to perform system call number 'callno', with parameters n1, n2, etc. The value returned in data register d0) is printed out.

Status of all processes

PS Gives lots of details of all processes operating.

Stop a background process

KILL process_name or PID Stop a particular background process.

Synchronise processes

"WAIT background_process ! action or command" &

Altering internal settings

OPTION optionnum setting
The option command is a general way of varying various fiddly settings within 1616/OS. If you use the option command without a second (setting) parameter, it will return the current setting of the option. Option *n* 0 usually turns the option on or off (opposite of the default).

option 0 1	(default) Turns on display of the current directory in the prompt.
option 1 1	Turns on verbose mode flag. Most commands operate quietly.
option 2 2	(default) Turns on alphabetic sorting. The option 2 setting also affects the sorting of wildcard expansion. option 2 0 turns off sorting of directory listings. option 2 1 turns on sorting of listings by date.
option 3 0	Affects how information is displayed when a machine exception occurs. If option 3 0 (default) has been selected then the screen is not cleared and a register dump only is displayed. If option 3 1 is selected the screen is cleared, the registers are dumped and a stack backtrace is displayed.

driver code.

option 5 0	(default) getmem returns a negative error code when out of memory. Option 5 1 causes the system to generate an internal error when the memory allocation function getmem receives a request for more memory than is available.
option 6 N	Sets the end-of-file character for character devices. The normal setting is .256 (none). Use 4 for Unix, .26 for MSDOS files. Reads from character devices terminate when this character is read. Like all options, Option 6 with no parameters prints out the current parameters, in this case EOF character.
option 7 8192	(default) Set `exec` file stack space.
option 8 0	(default) Disables the system from writing to the system blocks of a disk (blocks zero through to the start of the root directory). option 8 1 enables. Option is automattically turned off after use.
option 9 1	(default) Enable the Alt Ctrl R keyboard reset function.
option .10 1	(default) Enable the Alt S output suspension facility. option .10 0 disables. Note the period (.) indicating a decimal value,
option .11 1	(default) Enable all special keyboard Alt codes.
option .12 1	(default) Enables the output of a beep character when the system prints out an error message.
option .13 1	(default) Kill offending process upon bus or similar exception. option .13 0 is warm start.
option .14 1	(default) Only affects register dump in trace exception mode. Dumps contents of program counter and registers, if the 68000 trace flag is set on. You set the CPU trace flag in the 68000 status register with `or.w #$8000,sr`, and clear it using `and.w #$7fff,sr`. option .14 0 disables register output, except for PC.
option .15 0	Not available. Formerly set user ID number
option .16 .56	Not available. Formerly set `filemode` file creation mask to RWX.
option .17 1	Enable lower case pathnames in all files.
option .18 0	Set video output. 1 = ignore escape sequences. 2 = ignore meaning of control character. 3 = display **all** characters on screen. 4 = closer TVI 950 emulation.
option .19 1	Enable automatic re-read of all xpath directories, after encountering unrecognised program name. .19 2 searches xpath *before* current directory. .19 3 both

Quitting the command interpreter

QUIT [or End-of-File marker, usually Ctrl D, if set]

HANDY UTILITIES

Numeric base conversion

BASE n1 [n2] [n3] ... Into binary, octal, decimal and hexadecimal equivalents.

Entering the editor

EDIT filename1 [n1]
Full screen editor, ‘n1’ is the tab stop width (default is 8).

Assemb ling 68000 code

SSASM filename

Expression evaluation

EXPR n1 [op] [n2] [op] [n3] ...
If your first attempt at a calculation fails, remember last line recall!

x or X	multiplication
/	division
+	addition
-	subtraction
%	modulo (remainder)
&	bitwise AND
|	bitwise OR
^	bitwise exclusive OR

Printing the ASCII character set

ASCII [d | h | D | H] decimal or hex, 0-127, 128-255

Timing a command

TIME command How long did it take?

Defining function keys

FKEY n1 string1
Sets function key ‘n1’ to produce ‘string1’ when typed. String to be surrounded by quotes. Enter control keys by preceding them with Ctrl P). Get more than one command line by including the **Enter** (or Ctrl M).

Also direct from keyboard using Alt Ctrl function key combination.

COMMUNICATION

Reprogramming the serial ports

SERIAL channel baudrate rxbits txbits parity stopbits
0 is no parity, 1 is odd, 2 is even, 0 is 1 stopbit, 1 is 1.5, 2 is 2.

Using the 1616 as a terminal

TERMA
TERMB

Downloading S-records

SREC [filename] <redirection

7 Escape sequences

If you embed terminal control characters or escape sequences in a file, you can display **bold**, underline, *italics*, $_{\text{subscript}}$, and $^{\text{superscript}}$ text (or any reasonable mixture). The $29.95 *Dr Doc* editor uses these, as do other programs.

^G	Beep speaker 7.
^I	Tab 8.
^J	Line feed 10.
^K	Cursor up 11.
^L	Clear screen 12.
^M	Carriage return 13.
^V	Cursor down 22.
^^	Home cursor 30.
ESC =	(row+32) (col+32) Positions the cursor.
ESC)	Start highlighting.
ESC (	End highlighting.
ESC *	Clear the screen, or current window.
ESC B	(value+32) Sets the background colour to ‘value’.
ESC b	Visible bell.
ESC E	Insert a line at the current one.
ESC F	(value+32) Sets the foreground colour to ‘value’.
ESC G 1	Sets subscript mode.
ESC G 2	Sets superscript mode.
ESC G 4	Sets bold mode.
ESC G 8	Sets underline mode.
ESC G @	Sets italic mode.
ESC G 0	Clears subscript, superscript, underline, bold and italic modes.
ESC I	Back tab.
ESC j	Reverse scroll display.
ESC M	(from+32) (to+32) Copies the contents of a line.
ESC P	(position+32) (value+32) Writes ‘value’ into the 1616 video pallette at ‘position’.
ESC Q	Character insert.
ESC q	Enter insert mode.
ESC r	End insert mode. ($1B, $72).
ESC R	Delete the current line.
ESC S	(value+32) Sets the border colour to ‘value’.
ESC t	Clears from cursor to the end of the line.
ESC T	Clears from cursor to the end of the line.
ESC W	Delete character.
ESC Y	Clears from cursor to the end of the screen.

8 System calls

The Applix system calls are relatively low level functions in the EPROMS that can be accessed from the keyboard, or under program control. A full description is in the *Programmers Manual*. Invoke from the keyboard using `syscall .number`, followed by any required parameters. Remember that keyboard input defaults to hexadecimal values, so place a `.` before each number to indicate a decimal value. In assembler or C, is is easier to use the system call header files. A system call is actually performed by placing the call number in register D0, and any required arguments as 32 bit long integers into D1, D2, A0, A1 and A2, and then executing a 68000 Trap 7 instruction. D0 contains the negative error code, or other results, upon return.

General system calls

101	coldboot( )	Cold start 1616/OS
0	warmboot( )	Warm start 1616/OS
1	warmboot( )	Warm start 1616/OS
11	loadrel(ifd, addr)	Load relocatable code from open file
13	exit(exit_status)	Terminate transient program
16	set_vsvec(vec, rate, callval)	Install a vertical sync interrupt routine
17	clr_vsvec(vnum)	Remove a vertical sync interrupt routine
18	get_ticks( )	Get number of ticks since system startup
19	get_cpu( )	Determine CPU type
21	caswraw(start, length, leader)	Raw cassette block write
22	casrraw(buf, leader, maxhunk)	Raw cassette block read
23	getdate(buffer)	Get system time date
24	setdate(buffer)	Set system time date
25	abortstat( )	Get ALT-C status
26	ent1ints(vec, preload)	Enable VIA timer1 interrupts
27	dist1ints( )	Disable VIA timer1 interrupts
28	sine(angle)	Calculate a sine
29	def_fk(fknum, str)	Define a function key
30	getrand( )	Read random number seed
62	getmem(nbytes, mode)	Allocate memory
63	getfmem(addr, nbytes, mode)	Request memory at fixed address
64	freemem(addr)	Free memory
69	floadrel(path, memmode)	Load an executable file
80	setstvec(vecnum, whereto)	Alter/install a system call vector
83	gettdstr(buffer)	Get time/date string
83	cvttdstr(0, 0, dateptr, mybuf)	
83	readtimeinc(0, 1, 0)	
83	settimeinc(inc)	
84	nledit(str, length)	Edit a line with length constraint
86	ledit(str)	Line editor
87	iexec(prompt)	Indefinitely call 1616/OS command executor
88	exec(lb)	Execute a 1616/OS command
89	callmrd(n, cmd, arg)	Call memory resident driver
90	set_kvec(vec)	Alter keyboard scan code vector
91	clparse(pargs, ptype, pval)	Internal command parser
92	qsort(base, nel, width, compar)	General purpose sorting function
93	sliceargs(string, argv, wcexp)	String separation and wildcard expansion
94	cpuspeed( )	Return CPU speed
97	execa(argv)	Execute a system command
98	execv(path, argv)	Execute a system command
99	option(opnum, setting)	Set internal mode
122	errmes(errcode)	Interpret error code
126	getromver( )	Return 1616/OS version
128	aexeca(argv isasync)	Execute a system command, for multitasking
135	alias(cmd, arg)	Alias a command
142	fnledit(buf, len, in, out)	line edit remote

File I/O system calls

65	chdir(path)	Change current directory
66	mkdir(path, ndirblocks)	Create a new directory
67	getfullpath(path, memmode)	Get full pathname from relative pathname
68	pathcmp(path1, path2)	Compare two pathnames
100	inst_bdvr(br, bw, misc, name, bitmap)	Install a block device driver
102	find_bdvr(name)	Locate a block device driver
103	blkread(blk, buf, dev)	Raw block read
104	blkwrite(blk, buf, dev)	Raw block write
105	open(name, mode)	Open a disk file / character device
106	read(handle, buf, nbytes)	Read from a disk file / character device
107	close(handle)	Close a disk file
108	creat(name, type, addr)	Create a disk file
109	write(handle, buf, nbytes)	Write data to a disk file / character device
110	unlink(name)	Delete a disk file
111	rename(old, new)	Rename a disk file
112	filestat(name, buf)	Get the status of a disk file
113	readdir(dev, buf, dp, pos, pd)	Sequentially read disk directory
114	interpbec(ec, buf)	Interpret a block device error code
115	seek(fd, offset, mode)	Seek to a new disk file position
116	tell(fd)	Return current disk file position
117	bdmisc(bdnum, code, arg1)	Call block driver miscellaneous function
118	processdir(path, buffer, mode)	Perform directory operations
119	multiblkio(drv, cmd, ptr, blk, nblks)	Multiple block I/O driver
124	rdalldir(p, mm, sm, ps)	Read all directories
134	slink(one, two, three)	Symbolic link code
141	chkperm(pdirent, mask, path)	Check directory permissions

Character I/O system calls

2	getchar( )	Read one character from standard input
3	sgetchar( )	Get status of standard input
4	putchar(ch)	Put a character to standard output
5	sputchar( )	Get status of standard output
6	getc(dvr)	Get a character from a stream
7	sgetc(dvr)	Get status of an input stream
8	putc(dvr, ch)	Put a character to an output stream
9	sputc(dvr)	Get status of an output stream
14	set_sip(dvr)	Assign standard input
15	set_sop(dvr)	Assign standard output
20	set_ser(dvr)	Assign standard error
95	find_driver(ioro, name)	Locate a character device driver
10	add_ipdvr(io, stat, name, pv)	Install an input character device driver
10	add_xipdvr(io, stat, name, pv, m)	Install extended input drive
12	add_opdvr(io, stat, name, pv)	Install an output character device driver
96	get_dvrlist(ioro)	Locate the character device driver table
81	new_cbuf(dev, addr, len)	Vary buffer size for a character device
48	printf(str, p1, p2, p3, p4)	Formatted output
49	sprintf(buf, contstr, p1, p2, p3, p4)	Data formatting
120	fprintf(handle, contstr, p1, p2, p3)	Formatted output to a character stream
121	fputs(fd, buf)	String output to a character stream
123	fgets(f, buf)	Line input from a character stream
133	cdmisc(dvrnum, cmd, arg1, arg2, arg3)	misc character driver

---

Video output system calls

31	set_640(mode)	Set/clear 640 column mode
32	set_vdp(page)	Set video display page
33	set_vap(page)	Set video software access page

35		set_bgcol(colmask)	Set video text background colour mask
36		set_bdcol(col)	Set video border colour
37		set_pal(palpos, col)	Set a pallette entry
38		rdch_shape(charno)	Get pointer to character shape definition
39		def_chshape(charno, defptr)	Redefine a character shape
40		def_wind(wind)	Define a video window
41		vid_address(x, y)	Get physical video addresses
42		move_wind(buf, mode)	Move video window contents
43		rawvid(row, col, ch, fgmask, bgmask)	Low level video character drawing
44		fill_wind(col)	Fill the video window
45		scurs_mode(rate, enable, mask)	Alter cursor mode
46		mousetrap(trapno, vector)	Install mouse driver intercept
46		savecontext(4, p)	Save context
46		restvcontext(5, p)	Restore context
46		vcontextsize	Context size
130		newchset(ptr)	Select a new video character set

Graphics system calls

47	fill(x, y, value)	Area fill routine
50	rset_pel(x, y, val)	Raw graphics point draw
51	set_pel(x, y, val)	Windowed graphics point draw
52	rline(x0, y0, x1, y1)	Raw graphics line draw
53	drawline(x0, y0, x1, y1)	Windowed graphics line draw
54	rread_pel(x, y)	Raw pixel read
55	read_pel(x, y)	Windowed pixel read
56	sgfgcol(colour)	Set graphics foreground colour
59	rcircle(x, y, radius)	Raw circle draw
60	circle(x, y, radius)	Windowed circle draw
61	sdotmode(mode)	Set graphics write operation

OS control and Multitasking

119		multiblkio(drv, cmd, ptr, blk, nblks)	Multiple block I/O driver
125		oscontrol(cmd, arg)	System alterations
125	0	setibcvec(vec)	command disable
125	1	obramstart( )	start of on-board ram
125	2		not defined
125	3	readibcvec( )	command disable vector
125	4	forcelevel0( )	next reset is level 0
125	5	readxpath(n)	points to xpath name
125	6	readassign(m)	read back assigns
125	7	setumask(n)	set file creation mask for process
125	8	readumask()	read file creation mask
125	9	setuid(n)	set UID for process
125	10	readuid( )	read UID for process
125	11	get_bdev(path)	block devie driver no for file
125	12	dumplastlines(cdev)	throw away last lines for line editor
125	13	setwildvec(vec)	replace wildcard comparison
125	14	readwildvec( )	point to wildcard compare routne
125	15		not used
125	16		not used
125	17	video_init(level)	initialise video to any reset level
125	18	kb_init(level)	initialise keyboard to any reset level
125	19	setbeepvol(vol)	set volume of beep
125	20	readbeepvol( )	read volume of beep
125	21	setbeepvec(vec)	set sound vector for beep
125	22	readbeepvec( )	read sound vector for beep
125	23	nouseffbs(n)	don’t use 4 kbyte file patch
125	24	readffbs()	find state of 4 kbyte file patch

125	26	rxtxptr(n)	pointer to SCC ISR vector
125	27	setbdlockin(n)	lockin multiio call on devic
125	28	readbdlockin( )	read lockin state
125	29	startofday( )	non-zero if power up
125	30	pfastchar( )	fast video character output
125	31	srbrclock(n)	baud rate multiplier
125	32	timer1used( )	true if in freetone
125	33	trashassign(uid)	trash assigns by UID
125	34	trashenvstrings(pid)	trash environments by UID
125	35	envsubs(s, dollaronly, mmode)	string substitution mode
125	36	doassign(argc, argv)	access to assign command
127		lockZ80(mode, arg)	Lock Z80, file system interlock
128		aexeca(argv, isasync)	Multitasking execution of system command
129		proccntl(mode, arg(s))	Multiple command process control call
129	0	getpid	Current process ID
129	1	getppid(pid)	Get parent process ID
129	2	exit2(exitcode)	Terminate current process
129	3	kill(pid)	Unconditionally terminate process
129	4	sleep(pid)	Suspend processing
129	5	getproctab(pid)	Process table entry pointer
129	6	cwd(pid, path)	Change current working directory
129	7	lockin(mode)	Disable process descheduling
129	8	runstats(mode)	Collect statistics of stack & memory useage
129	9	wait(pid)	Wait for process to exit
129	10	nice(pid, ticks)	Set process time slice
129	11	sigsend(pid, arg1, arg2)	Send a signal
129	12	sigcatch(vector)	Install a signal handler
129	13	sendinterrupt(rootpid)	Send Alt-Ctrl-C interrupt
129	14	proctrace(rootpid)	Put process into trace mode
129	15	getprocflags(pid)	Read process flags from process table
129	16	isinteractive(pid)	Is process interactive?
129	17	nospcheck(pid, mode)	Enable and disable stack checking
129	18	csvec(v)	Install context switch vector
129	19	getpcurpid( )	Pointer to current PID
129	20	readsigvecpid)	Signal handler for current process
129	21	fsbptr( )	File system interlock semaphore
129	22	fspptr( )	PID of process in file system
129	23	ssptr( )	Lock current process in file system
129	24	killuser(hs)	Kill processes of homeshell PID
129	25	sigblock(pid, blk)	Block or unblock signals
129	26	alarm(n)	Set an alarm
129	27	sigblocker(rootpid, sig, arg)	Signals blocked process
129	28	snooze(vec, p1, p2)	Sleep until true
129	29	siguser(pid, sig, arg)	Signal user
129	30	findhomeshell(pid)	Find homeshell of process
129	31	setshnice(pid, nice)	Set nice level of shell
129	32	lastchild(pid)	Find most recent child of PID
129	33	swpptr( )	Switch pending flag
129	34	killdown(startpid)	Send killdown signal to parent
129	35	sigdown(statpid, val1, val2)	Send sigdown signal to parent
129	36	killuid(uid)	Kill all processes of UID
129	37	siguid(uid, p1, p2)	Signal all proceses of UID
129	38	setsigexit(pid, mode)	Signal process on other exit
129	39	setpg(pg)	Set up a process group
129	40	sigpg(pg, sig, arg)	Signal a process group
129	41	killpg(pg)	Kill a process group
129	42	setprocumask(id, mask)	Set a process group umask
129	43	setenvbits(pid, mask, set)	Set environment bits for a user
129	44	getenvbits(pid)	Find which environment bits are set
129	45	nametopid(pid)	Convert string to PID

129	47	blockrx(mode)	Receive an interprocess block
129	48	setenv(pid, name, set, mode)	Manipulate environment strings (set)
129	49	getenv(pid, name, mode)	Get environment string variables
131		schedprep(addr, argv, flags, ss)	Schedule a new process
132		pipe(ptr)	Pipe

Hardware control system calls

70	anipsel(ipnum)	Select an analogue input
71	anopsel(opnum)	Select an analogue output
72	anopdis( )	Disable analogue outputs
73	adc( )	Perform analogue to digital conversion
74	dac(val)	Perform digital to analogue conversion
75	set_led(val)	Set/clear LED
76	freetone(table, tablen, length, preload)	Play a waveform
77	fttime( )	Return time left for freetone completion
78	rdiport( )	Read input port
79	rdbiport( )	Read time-accumulated input port
82	prog_sio(chan, spptr)	Reprogram a serial port
133	cdmisc(dvr, cmd, p1, p2, p3)	Misc char device driver
140	crtc_init(mode, p)	Initialise CRTC mode

9 Available Programs

This lists the command line options for many of the programs available on the Applix 1616, and briefly describes the programs. Source code for programs accompanies the program (except for `sc` spreadsheet calculator, which uses curses heavily). Many utilities are available on the $29.95 *Utilities* disk, others come with the disk drive kit, the rest are available on 25 shareware disks, available at $5 each, plus $2 postage.

| or ^ **pipe** [**on**] [**off**] [**d**] UNIX style memory resident pipe. *U1am*
1616typer.bas Very simple letter writer, justifies typing. *4pc*
3demo 3 D demo, converted from Atari ST. *18gb*
60hzcrtc Modify 6545 CRTC video controller registers. *6am*
640circle Makes circles display correctly in BASIC. *4am*
adc `filesize filename` capture sound 38.97 kHz, compresed storage. *19gb*
adc11 `soundfile` Input sound from external digitiser using 6522 VIA. *17ms*
adc11 `tablesize soundfile` Input sound from microphone via user port. *18gb*
adc15 `tablesize soundfile` Input sound faster from microphone via user port. *18gb*
adc22 `tablesize soundfile` *gb*
adc_s `tablesize soundfile` Input stereo from two microphones via user port. *18gb*
addcr `infile outfile` Add carriage returns to linefeeds in file. *U1am*
addcr `filename` Add CR to LF in file, for editing. *20am*
addlf `filename` Add LF to CR in file, for editing. *20am*
advent [`-red`] Colossal Cave text adventure game. *2mh.am*
ampl Sound sampling, no hardware. Based on Gerhard's. *24am*
amv.bas Animal vegetable mineral guessing game. *4df*
analogout Demo use of analog ports, selection. *25*
app Part of Conal' ega mrd for OSv4. *18cw*
ar [**-adprtvx**] `afile` [`filename`] ... File archive and library. *U1am*
arc [**-**]{`amufdxeplvtc`} [`biswn`] [`g`*`password`*] *`archive`* [*`filename ...`*] File archive and library, with compression. *U1am*
arep `repetition_ticks initial_delay_ticks` Keyboard autorepeat and delay. *Uam*
arep`.mrd` MRD for keyboard autorepeat rate and delay. *24am*
asciicalc ascii string evaluation tutorial. *6sy*
atoz `filename` Transfer Applix file to CP/M format. *11jm*
back `soundfile` Play sound file backwards. *1,3jf*
back11 `soundfile` Play sound file backwards. *1,6jf*
back22 `soundfile` Play sound file backwards. *1,6jf*
ball Ping pong video ball game. *3ad*
basic Public domain Tiny BASIC interpreter. *14am*
beep.mrd MRD to change ^G beep sound. *23amc*
bigbuf `[bufsize] [satx] [sarx] [sbtx] [sbrx] [cent] [kb]` buffer size *14am*
bkg Backgammon game against computer. *2mh.am*
blockdev `devname` Initialise, format and name disks. *Uam*
bmedit `bmfile` Bit map editor for making icons and characters. *15amc*
brickwalls.bas Manoeuvering video game. *4df*
bs [**-b**|**-s**][**-c**] Curses based battleship game, from Unix. *18am*
bug & Demonstration of mouse cursor. *Uam*
buildmrd [`-r`*`ramdisksize`*] [`-o`*`outfile`*] [`-s`*`stacksize`*] [`-v`*`videosize`*] [`-c`*`colours`*] `[file.xrel] [file.xrel] ...` Establish TSR (MRD) programs, configure memory. *Uam*
c HiTech commercial C compiler, *Clyde Smith-Stubbs*, $275.00
c1616 `helpfile` [`swapfile`] Hypertext help system. *13cw*
`c_examples` Simple C examples. *18dw*
cfront C++ front end. *25cm*
cal [`month`] `year` Any year, any month calendar. *U1am*
calendar.bas Calendar in BASIC. *4pc*
cdev [`device:`] Dump character device control tables. *15amc*
chess GNU chess program, with graphics interface. *11am*
chmem `stackspace files ...` Change file stackspace usage. *Uam*
choice -s -h Two handed trump taking card game. *18am*
chop Code execution profiler, for hacking code to go faster. *24am*

clrwin Tutorial on clearing a window. *6sy*
cmp [-**ls**] file1 file2 Compare contents of two files. *U1am*
colour-bars Test your display colours. *5mj*
comm [- [**123**]] file1 file2 Find lines common to two files. *U1am*
convert filein fileout Amiga sound files to 1616. *15mg*
cproto Generate function prototype header from .c file. *24am*
crobots file file Robot strategy in C. *16cm*
crtsav [time] Screen saver, blanks display after set time. *3jf*
d_list and **dlist2** directory lister for entire disk. *18gb*
dateset Easy way to set system date and time. *5mj*
dateset.bas Set system date and time. *4df*
dd [option=value] ... Convert, translate and copy files. *U1am*
debug include in your code to trace. *14gb*
demonstration 68000 exception handler tutorial. *5mj*
df filename [-**n***nnnn*] [-**o***nnnn*] Dump contents of disk file. *U1am*
dhrynoreg dhryreg Traditional benchmarks for CPU and C compiler. *9am*
dial phone Hayes modem autodialler utility. *15mg*
diff file1 file2 Differential file comparator. *U1am*
dis filename.exec Early version 68000 file disassembler. *U1am*
dis filename.xrel 68000 file disassembler. *U1am*
dis start address count Disassemble 68000 memory. *U1am*
disassemble file.xrel [char] Another disassembler. *14gb*
disassemble m address1 address2 [char] Disassembler memory. *14gb*
diskcopy sourcedev destdev [-f] [-r] [-s] [-v] Smart disk copy utility. *Uam*
diskio Disk input output tutorial. *6sy*
diskstat Disk statistics. *24am*
dissolve Randomly clear video display, special effect. *6ch*
doc-write Menu front end and file handler for editor. *16mj*
dosdir [**-lr**] drive [dir] List MS-DOS disk directory contents. *7gs*
dosget [**-a**] drive file1 [file2] ... Reads MS-DOS disk files, writes them to current Applix drive. *7gs*
dosinit drive Initialise an empty MS-DOS file system on formatted disk. *7gs*
dosput [**-a**] drive file [file2 [file3 ...]] Write files to MS-DOS drive. *7gs*
dosread [**-a**] drive file Display a file from MS-DOS disk. *7gs*
doswrite [**-a**] drive file Write a file on MS-DOS disk, from standard input. *7gs*
dr doc Document editor, fancy screens, commercial $29.95, *am*
driveparm [drive [steprate doublestep bitmap sizecode sectors tracks sides cache diskchange]] Display or change all disk drive parameters, for alien disks. *7gs*
drivetime drive Time drive rotation rate. *7gs*
du2 Select and view raw disk blocks. *6sy*
du Disk usage reporter, like Unix. *24am*
dumpbro Print graphics display to Brother M1109 printer. *6ms*
dumpch Display character shapes in hexadecimal. *6sy*
dumpscreen Print graphics display to Epson printer. *3nc*
easy_write Menu front end and file handler for compiler or assembler. *15mj*
ega.mrd Part of conal's ega drivers for OS V4. *18cw*
ellipse Draw ellipse on display. *24am*
emacs Traditional minicomputer text editor. *14cm*
err number Error message number translator. *23amc*
ep232 Drive smart EPROM burner from RS232 port. *3jm*
eroff [*+00*] [*-00*] [-**s**] [-**h**] [files] Enhanced version of roff text formatter. *U1am*
except.mrd 68000 exception handler, to help trace bugs. *3,5mj*
exp arguments Math and logic expressions evaluated. *U1am*
factorial [**-v**] number Calculates factorials to many places. *15gb*
fastcopy source dest Set drive characteristics, will quickly copy **any** non-Macintosh alien disk, such as IBM, Applix, Atari. *7gs*
fft Fast Fourier Transform code sample. *9cm*
fileclean filename Converts imported files to editor format. *Uam*
find dirspec filespec Find a named file on disk. *U1am*
float Floating point routines in assembler, for fractals. *19gb*

fmem Free memory statistics, various modes. *20cd*
forever command Repeat any command until cancelled. *Uam*
format filename [*n*] Dot matrix printer utility. *3db*
format drive Universal disk format program (disk will not work until a file system is also placed on it - use blockdev for Applix, mkfs for Minix, dosinit for MS-DOS, etc.) *7gs*
frac datafile Pretty fractal pictures. *6sn*
fscheck devname **[-v] [-y] [-yy] [-q]** Check and repair disk file system. *Uam*
fscopy sourcedir destdir Copy directory and all sub-directories. *U1am*
ft etc Freetone sound effects, various types. *16sy*
ftolower < infile > outfile Convert file contents to lower case. *U1am*
ftoupper < infile > outfile Convert file contents to upper case. *U1am*
ftree [**-s** filename] [**-d**] [**-f**] [/dev] Find and display files and directories. *15mj*
gbplay playspeed soundfile V1.3 of Gerhard's sound file player. *19gb*
genreloc fname1 fname2 ofname [bsssize] Make program relocatable. *Uam*
gensrec infile > output Output in Motorola S-record format. *U1am*
getty [sb:] [**-u***id*] [**-m***alrwx*] [**-c***barfile*] Restricted shell via serial ports. *15amc*
gp pattern file Search a file for a simple pattern. *U1am*
grav Bouncing ball demo converted from MGR. *24am*
greed [**-s**] Eat numbered points game, very addictive. *16am*
grep [**-cfinv**] pattern [file ...] Search files for complex pattern. *U1am*
guts startline [endline] file ... Output lines from file. *19dw*
hdbackup /hdrive /fdrive Back up hard disk. *U1am*
hdrestore /sourcedrive /destdrive Restore hard disk from backup. *U1am*
head [**-n**] file1 [file2 ...] Show first n lines of files. *U1am*
hexagon Block the opponent peg game. *5mj*
HItech **C** compiler, commercial, $275.00
hs name *or* number Early version programmers help system. *13cw*
ibmfont Use IBM graphics font instead of inbuilt one. *Uam*
imagewriter filename.roff Printer utility for runoff files. *10cmills*
indent [input-file [output-file]] [-bad | -nbad] [-bap | -nbap] [-bbb | -nbbb] [-bc | -nbc] [-bl] [-br] [-cn] [-cdn] [-cdb | -ncdb] [-ce | -nce] [-cin] [-clin] [-dn] [-din] [-fc1 | -nfc1] [-in] [-ip | -nip] [-ln] [-lcn] [-lp | -nlp] [-pcs | -npcs] [-npro] [-psl | -npsl] [-sc | -nsc] [-sob | -nsob] [-st] [-troff] [-v | -nv] Pretty up your C source files. *10am*
infocom filename Guide to Infocom text adventures. *6cm*
install *24am*
invertch Invert the normal character set. *6sy*
julia Draws Julia set fractals. *19gb*
justify filename Justify a text file for printing. *3nc*
kal Kaleidoscope graphics. *1jf*
karma Collect happiness, convert the neighbourhood game. *4pa*
kmem C storage allocator. *6am*
knuth_rand Notes on C random number generator in C. *24*
kv Print keyboard scan codes. *6am*
laser Futuristic reflecting battle chess game. *4pa*
ledit Line editor example. *24am*
lfk fkey_def_file Define function keys more easily. *15ch*
lhx - Lharc arc utility. *23amc*
lib File organiser, indexes files by disk. Very handy. *19gb*
lib File organiser, indexes files by disk. Very handy. *20gb*
lif Conway's Game of Life in 68000 assembler. *14gb*
life Conway's Game of Life in C. *3,5mj*
life Yet another life. Source, no xrel. *23amc*
linkmrd.mrd MRD required for **ls**. *20jf*
lisp Version 1.6 of old AI language. *8am*
load4000 filename Place relocatable program in default memory location. *U1am*
loadreloc ifname.xrel Place relocatable program in fixed memory location. *Uam*
locate.mrd Find MRD (TSR) and other programs and data in memory. *15mj*
ls - Unix like ls directory command, needs linkmrd. *20jf*

macpic filename ... V4.1 new, updated. *19amc*
macshow filename Convert, colour and view Macintosh pictures (early version). *1amc*
make [**-f** makefile] [**-dinpqrst**] [macro=val] [target(s)] [names] Aid in compiling C programs. *U1am*
makecmd cmdfile.cmd start1 end1 [...] entry Create a TRSDOS format .cmd file from Z80 code in disk controller. Used for alien disks. *7gs*
makeega Adjust 6545 CRTC to run EGA video on Hercules monitor. *15lw*
mand Mandelbrot demonstration, same as in Forth package. *14gb*
mand Mandelbrot demo, two versions, similar to above. *18gb*
mandel Mandelbrot demo, updated. *19gb*
mastermind.bas Traditional number guessing game. *4df*
maze Obscure C competition maze *24*
mcd [pidname] newdir Change MGR's working directory. *18am*
mem Report of available free memory, largest free block, etc. *15amc*
mem logfile Memory usage daemon, run in background. *Uam*
mexec [**-v**] printf_control_string [args] Execute any command, in all directories. *U1am*
minix multitasking Unix workalike, ported by *Colin McCormack*, commercial, $225.00
mkcd dirname Make dir and change to it. *24am*
mkfloppy Shell demo of << command. *24am*
mkproto Generate function prototype header from .c file. *24am*
modem sa: [sb:] Xmodem file transfer program. *1,2mh*
modem32 Menu driven file transfer program. *6sy*
more [files ...] Shows files a page at a time. *U1am*
more [files ...] Shows files a page at a time, goes backwards, searches. *15mgrd*
mrdstat Shows details of MRD (TSR) programs in memory. *U1am*
multiply. Source demo of unsigned 68000 long multiply. *24*
mvs file Convert file to total confusion. *25ch*
names [number] Generate alien sounding names. *14am*
newbeep.mrd MRD to allow selectable beep sound. also selbeep. *24am*
nim.bas Traditional math logic game. *4df*
nswp Bulk file copy, delete, view utility, like simple Xtree. *6sy*
nzdebug File dump and edit, like IBM debug. *19dw* or *gb*?
ole Test for floating point bug in C compiler. *19on*
para Paranoia text adventure, always gets you. *2amc*
pcrip Shoot the IBM game, for joystick. *Uam*
pfile.bas Simple data base, includes index of last decade of *Electronics Australia*. *4pc*
pic soundfile Visual display of sound files. *18gb*
| or ^ **pipe** [**on**] [**off**] [**d**] UNIX style memory resident pipe. *U1am*
pipe & Simulates serial link, for test purposes. *24am*
play [freq] soundfile [soundfile ...] Play digitised soundfiles at selected frequency. *9gb* also *18gb*
play_s playspeed soundfile Play digitised sound in stereo. *18gb*
playsound soundfile Early version. *1am*
pr [+*page*] [-*columns*] [**-h** header] [**-w** *width*] [**-l** *length*] [**-bfnpst**] [files] Text format and print routine. *U1am*
prog? 68000 assembler tutorials. *1km*
ps2 Variation on process status command. *Uam*
ps3 Another variation on process status command. *15amc*
pwm & Pulse width modulate LED (flash it). *19 ac am jf*
quick.mrd Reduce lines of video display to speed processing. *9am*
quindex Quick index, file menu and program launch system. *5mj*
quix Bouncing line demo, quick, memory board aware. *23amc*
rap Jive song generator (Andrew was in USA too long) *24am*
rawcopy sourcedevice destdevice Copy Minix file system disks. *Uam*
rawread /device address blocknum [blocknum] Read absolute disk blocks (carefully). *U1am*
rawwrite /device address blocknum [blocknum] Write absolute disk blocks. *U1am*
rb [-7buv] [-sA] [-sB] Obsolete yam and ymodem recieve routine. *3cm*
rb [-1bcuv] filename Obsolete yam and ymodem receive routine. *3cm*

readver Display the disk controller version number (alien disks). *7gs*
relcc [**-acdijlorsuv**] file C compiler preprocessor, produces relocatable code. *Uam*
renlower [file] [file] ... Rename filename on disk to lower case. *Uam*
renupper [file] [file] ... Rename filenames on disk to upper case. *Uam*
reverb soundfile ptr1_off ptr1_lvl prt2_off ptr2_lvl ptr2_off ptr3_lvl Apply 3 level reverberation to playing of soundfile. *1,6jf*
rmcr infile outfile Remove carriage returns from text file. *U1am*
rmlf filename Remove LF from file, for editing. See clean. *20am*
robot cprog1 cprog2 Write a robot in C, play against another. *16cm*
roff [*+00*] [*-00*] [**-s**] [**-h**] [files] Text file formatter and printer. *U1am*
runoff [+n] [-n] [-p] [-s] Expanded text file formatter. *10cmills*
rvuc Review C source code. *24am*
sa String art, converted from MGR. *24am*
sample Sound sampler, no hardware, based omn gerhard's. *24am*
sb [-7dfkquv] [-sA] [-sB] filename ... Obsolete ymodem send file. *6cm*
sc [-c] [-m] [-n] [-r] [-x] [file] Spreadsheet calculator, full maths and macros. *15am*
scan List keyboard scan codes. *9am*
scc Calculate Zilog 8530 SCC code constants. *U1am*
scc Calculate Zilog 8530 SCC code constants. *24am*
sdate Set date and time easily, pop up window. *6amc*
sdate [-h] Set date and time easily. *15mgrd*
sea - Sealink arc utility. *23amc*
sega.mrd Change video to EGA 640 by 400 mode. *6cw*
selbeep Select a beep sound, require newbeep.mrd. *24am*
setload filename address Change executable address of file. *U1am*
setstep drive stepcode Set the disk drive step rate, for alien disks. *7gs*
setvol name [device] Change volume name of diskette. *15mj*
shareware.dat List of all disk files to SW#20, for use with lib *20gb*
size filename[.xrel] ... Size of text, data, bss, stack for files. *18am*
solitare Jump the peg board game. *5mj*
sort [**-funbirdcmt**'x'] [*+pos* [*m.n*] [*-pos*]] [**-o** outfile] [files] Sort a file, alpha, reverse, by position, field, etc. *U1am*
sound-11 soundfile Play an 11 kHz digitised sound file. *3mg*
sound-22 soundfile Play a 22 kHz sound file. *3mg*
sound11 soundfile Play an 11 kHz sound file. *1am*
sound22 soundfile Play a 22 kHz sound file. *1am*
soundsloman soundfile Customised version, play a sound file. *17ms*
space-potatoes.bas They chase you game. *4df*
speakfile soundfile Easy version, play a sound file. *1am*
split [**-n**] [file [name]] Split an overlarge file into pieces. *U1am*
ssbasic BASIC for 1616, commercial, $69.00
ssdcerr errorno Disk error displayed in text, for alien disks. *7gs*
sseg Part of EGA 640 by 400 graphics package. *12cw*
ssforth Floating point Forth, by Peter Fletcher, commercial with source, $89.00
sstools File and directory manager, like commercial Xtree. Great. *19cmills*
star Logical board game. *5mj*
stat -b File/dir info, block load, stat bits, blockmap, etc. *18am*
sterm sa:|sb: Extended terminal program, drop DTR, break, load, rot13. *20jf*
strings [-minsize] [files] Find ASCII text strings in files. *U1am*
sum file Checksum a file. *u1am*
syllogy Logic game, demonstrates ai. *5mj*
sylloword Logic game, early version of syllogy. *5mj*
sync drive Flush read and write caches for alien disks. *7gs*
tail [-*number*] [file] List last n lines of files. *U1am*
tee [**-i**] [**-a**] [file] ... Pipe fitting, output to display and device. *U1am*
t flight simulator outline in assembler. *18gb*
tc-017 flight simulator outline, also in assembler. *18gb*
term chardev Better terminal program. *18am*
term chardev Can capture incoming to disk. No xrel. *24am*
terminal port [ver] [half] [inLF] [outLF] [print] [echo] [[-]logfile] Elaborate

termite Eats your display, special effect. *8amc*
test Test an MRD before installing. *5mj*
testmrd Test MRD (TSR) prior to installing. *5mj*
teststr14 `password` Password protection and screen blank. *16on*
tictactoe Traditional noughts and crosses game (never loses). *8am*
tiff description Description of suggested music file format. *10ls*
tinybasic Tiny BASIC, extensions, plus examples, manual. *19dw & mgrd*
toupper Filter to upper case. *15mgrd*
trace Single step a program. *3mgreg*
trace `address` Single step a program. *Uam*
tree [**-q**] [`/vol`] List all directories and files. *3mg*
tree [**-q**] [`device name`] List all directories and files. *3mg*
tstint Test whether program is interactive or background. *Uam*
tune3 `musicfile` Play 3 voice music files. *Uam*
music files, various *9cmills*
uid Display user ID number. *Uam*
uid *n* Change user ID number. *Uam*
uid *n* `files` Changes user ID on list of files. *Uam*
uid *n* `-` Changes user ID on files listed in standard input. *Uam*
umask -alrwx [-] [-v] [file] [file] ... Set user file permissions mask. *Uam*
unzip File compression program. *20jf*
vcon Video context switcher, multiple virtual consoles. *Uam*
vconp Video context switched, three windows on one display. *Uam*
undelete [`files ...`] Recover recently erased files. *U1am*
uniq [**-udc** [+*n*] [-*n*]] [`input` [`output`]] Eliminate multiple identical lines in file. *U1am*
vi UNIX visual editor, from Minix. *9cm*
video Needs to be installed as an MRD on boot, provides EGA. *6cw*
video.ega Part of Conal's ega mrd drivers. *18cw*
vtrek Star Trek adventure game, character graphics only. *1mh*
wanderer Graphics adventure game, treasure quest. *8am*
warship Depth bomb the submarine game. *3ad*
warship2 Revised depth bomb the submarine game. *16am*
wave `soundfile` Display soundfile on video and play. *6ms*
wc [**-lwc**] [`files`] Count lines, words and characters in file. *U1am*
whereis `file[.xrel]` Find executable program on disk. *18am*
wh `file` Faster find executable program on disk. *20jf*
whir.mrd MRD to toggle speaker on each syscall. Cute. *23amc*
window Demonstration of window code. *3cm*
wisdom `wisdom.txt wisdom.idx` Cute quotes for all purposes. *16am*
wisdom `oracle.txt oracle.idx` Different cute quotes. *20jf*
xreltoexec `[loadaddr] xrelfile` Loads xrel file as exec file. *18am*
xreltoexec `[loadaddr] xrelfile` Loads xrel file as exec file update. *24am*
ya.shell Complicated yahtzee dice game. *6am*
ymodem `sa: [sb:]` Ymodem file transfer routines. *8cm*
yow esponds with inspirational mesage from file. *19dw*
Z80asc Z80 memory map dump for disk controller. *6sy*
Z80hex Z80 memory map dump for disk controller. *6sy*
zd Transfer files between CP/M and Applix. *11jm*
zmdb `a1 a2 n1` **zmfb** `a1 a2 n1` **zmwb** `a1 n1 [n2 ...]` Read, fill and write to disk controller Z80 memory (much the same as 1616's 68000 monitor). *11jm*
zmodem File transfer protocol, xrels are _rz and _sz. *25cm*
zoo `-{acDeglLPTuUvx}[aAcCdEfInmMNoOpPqu1:/.@n] archivefile` Zoo archive utility, no source yet. Zoo h for help. *23amc*
zrdos CP/M system using ZCPR, commercial item $150.00. *Conal Walsh*
ztoa `filename` Transfer files from CP/M to Applix. *11jm*

The location and authors of all software is shown above in the *italic* text at the end of each program. A number indicates the shareware disk with that number. A capital U is a User Disk, usually the one with 1616/OS V4. A H indicates a Hard Drive User Disk, which contains additional programs mostly by Mark Harvey. A U1 indicates a disk from the $30 Utility Disk set.

Andrew Driver (ad), Andrew Morton (am), Andrew McNamara (amc), Craig Dewick (cd), Cameron Hutchison (ch), Colin McCormack (cm), Craig Mills (cmills), Conal Walsh (cw), Dale Barnes (db), Dave Fowler (df), Dave Wilson (dw), Gerhard Baumann (gb), Greyham Stoney (gs), Jeremy Fitzhardinge (jf), Joe Moschini (jm), Kathy Morton (km), Lindsay Scales (ls), Lindsay Washusen (lw), Matthew Geier (mg), Matthew Gardener (mgrd), Mark Harvey (mh), Michael Johnson (mj), Michael Sloman (ms), Norm Clark (nc), Ole Nielson (on), Peter Ashby (pa), Paul Cahill (pc), Stephen Nicholson (sn), Sid Young (sy).

Cursor movement

^E	Up one line	^X	Down one line
^QE	Up to top of page	^QX	Down to bottom of page
^R	Up one page	^C	Down one page
^QR	To start of file	^QC	To end of file
^S	Left one character	^D	Right one character
^A	Left one word	^F	Right one word
^QS	Left 80 characters	^QD	Right 80 characters
^B	Start/end of line	^J	Start of next line

Scrolling

^Z	Scroll up	^W	Scroll down

Text deleting

^H,BS	Delete char backward	^G, DEL	Delete char forward
^Y	Delete line	^T	Delete word forward
^V	Delete line backwards	^QY	Delete line forward

Block commands

^KB	Mark block start	^QB	Go to block start
^KK	Mark block end	^QK	Go to block end
^KY	Delete marked block	^KV	Move marked block to cursor
^KH	Hide block	^KC	Copy marked block to cursor
^KW	Write block to file	^KP	Put block in undo buffer

File commands

^KR	Read in (merge) a file	^KD	Write out file, continue editing
^KX	Write out file, quit		

Miscellaneous

^KQ	Exit without saving file	^KI	Escape to 1616/OS
^KE	Execute 1616/OS command	^QG	Go to line number
^K0	Set block markers 0 to 9	^KF	Partial screen freeze
^Q0	Go to a block marker	^QF	Find pattern
^QA	Substitute pattern	^L	Repeat last pattern find
^N	Repeat last substitution	^U*n*	Undo buffer n
Esc	Redraw screen		