

SYSTEM2

Conversion program

COLLABORATORS

	<i>TITLE :</i> SYSTEM2		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Conversion program	October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	SYSTEM2	1
1.1	Overview of SYSTEM2	1
1.2	SYSTEM2	1
1.3	SYSTEM2	1
1.4	SYSTEM2	2
1.5	SYSTEM2	2
1.6	SYSTEM2	2
1.7	SYSTEM2	2
1.8	SYSTEM2	3
1.9	SYSTEM2	3
1.10	SYSTEM2	3
1.11	SYSTEM2	3
1.12	SYSTEM2	3
1.13	SYSTEM2	4
1.14	SYSTEM2	4
1.15	SYSTEM2	4
1.16	SYSTEM2	4
1.17	SYSTEM2	4
1.18	SYSTEM2	5
1.19	SYSTEM2	5
1.20	SYSTEM2	5
1.21	SYSTEM2	5
1.22	SYSTEM2	5
1.23	SYSTEM2	6
1.24	SYSTEM2	6
1.25	SYSTEM2	6
1.26	SYSTEM2	6
1.27	SYSTEM2	7
1.28	SYSTEM2	7
1.29	SYSTEM2	7

1.30 SYSTEM2	7
1.31 SYSTEM2	7
1.32 SYSTEM2	8
1.33 SYSTEM2	8
1.34 SYSTEM2	8
1.35 SYSTEM2	8
1.36 SYSTEM2	8
1.37 SYSTEM2	9
1.38 SYSTEM2	9
1.39 SYSTEM2	9
1.40 SYSTEM2	9
1.41 SYSTEM2	9
1.42 SYSTEM2	10
1.43 SYSTEM2	10
1.44 SYSTEM2	10
1.45 SYSTEM2	10
1.46 SYSTEM2	11
1.47 SYSTEM2	11
1.48 SYSTEM2	11
1.49 SYSTEM2	11
1.50 SYSTEM2	11
1.51 SYSTEM2	12
1.52 SYSTEM2	12
1.53 SYSTEM2	12
1.54 SYSTEM2	12
1.55 SYSTEM2	12
1.56 SYSTEM2	13
1.57 SYSTEM2	13
1.58 SYSTEM2	13
1.59 SYSTEM2	13
1.60 SYSTEM2	14
1.61 SYSTEM2	14
1.62 SYSTEM2	14
1.63 SYSTEM2	14
1.64 SYSTEM2	15
1.65 SYSTEM2	15
1.66 SYSTEM2	15
1.67 SYSTEM2	16
1.68 SYSTEM2	16

1.69 SYSTEM2	16
1.70 SYSTEM2	16
1.71 SYSTEM2	17
1.72 SYSTEM2	17
1.73 SYSTEM2	17
1.74 SYSTEM2	17
1.75 SYSTEM2	18
1.76 SYSTEM2	18
1.77 SYSTEM2	18
1.78 SYSTEM2	19
1.79 SYSTEM2	19
1.80 SYSTEM2	19
1.81 SYSTEM2	19
1.82 SYSTEM2	19
1.83 ;name	20
1.84 SYSTEM2	20
1.85 SYSTEM2	20

Chapter 1

SYSTEM2

1.1 Overview of SYSTEM2

Overview

An Acid Software Library

Converted to AmigaGuide by

Red When Excited Ltd

Used with the permission of Acid Software

Edited, fixed and cleaned by Toby Zuijdveld 27/02/1999.
mailto:hotcakes@abacus.net.au

1.2 SYSTEM2

Statement: MOVEQ

Modes :

Syntax : MOVEQ Direct value,D0-7

Transfer a 8 bit constant to data register D0-7 quickly

1.3 SYSTEM2

Statement: MOVEM

Modes :

Syntax : MOVEM Registers,Address or Register

Transfer several registers at the same time

Syntax : MOVEM Address or Register, Registers

Transfer several registers at the same time

1.4 SYSTEM2

Statement: MULS

Modes :

Syntax : MULS Source operand, Destination operand

Sign correct multiplications of two words to a long word

1.5 SYSTEM2

Statement: MULU

Modes :

Syntax : MULU Source operand, Destination operand

Multiplication without sign, similar to MULS

1.6 SYSTEM2

Statement: NBCD

Modes :

Syntax : NBCD Source operand, Destination operand

Negate a BCD number (Nine's complement)

1.7 SYSTEM2

Statement: NEG

Modes :

Syntax : NEG Address or Register

Negate an operator (Two's complement)

1.8 SYSTEM2

Statement: NEGX

Modes :
Syntax : NEGX Address or Register

Negates an operator with transfer

1.9 SYSTEM2

Statement: NOP

Modes :
Syntax : NOP

No Operation

1.10 SYSTEM2

Statement: NOT

Modes :
Syntax : NOT Address or Register

Inverts an operand (0s become 1s and vice versa)

1.11 SYSTEM2

Statement: OR

Modes :
Syntax : OR Source operand, Destination operand

Logical OR

1.12 SYSTEM2

Statement: ORI

Modes :
Syntax : ORI Direct value, Address or Register

Logical OR with a constant

1.13 SYSTEM2

Statement: PEA

Modes :
Syntax : PEA Address or Register

Put an address on the stack

1.14 SYSTEM2

Statement: rese

Modes :
Syntax : rese

Reset peripheral device (carefull!)

1.15 SYSTEM2

Statement: ROL

Modes :
Syntax : ROL Direct value,Address or Register

Rotate left

1.16 SYSTEM2

Statement: ROR

Modes :
Syntax : ROR Direct value,Address or Register

Rotate right

1.17 SYSTEM2

Statement: ROXL

Modes :
Syntax : ROXL Direct value,Address or Register

Rotate left with transfer to X flag

1.18 SYSTEM2

Statement: ROXR

Modes :

Syntax : ROXR Direct value,Address or Register

Rotate right with transfer to X flag

1.19 SYSTEM2

Statement: RTE

Modes :

Syntax : RTE

Return from an Exception

Let's look at the TRAP commands. They aren't used in the Amiga operating system. A TRAP command and a number between zero and fifteen are used to call one of the 16 possible TRAP routines. If the command TRAP #0 is executed, the processor (in Supervisor mode) branches to the routine whose address lies at \$80 in memory. This routine must end with a RTE (ReTurn from Exception) command.

1.20 SYSTEM2

Statement: RTR

Modes :

Syntax : RTR

Return and load flags

1.21 SYSTEM2

Statement: RTS

Modes :

Syntax : RTS

Return from a subroutine (after a BSR or JSR command)

1.22 SYSTEM2

Statement: SBCD

Modes :

Syntax : SBCD Source operand, Destination operand

Subtract two BCD coded numbers

1.23 SYSTEM2

Statement: SCC

Modes :

Syntax : SCC Address or Register

Set a byte to -1 if the condition is fulfilled

1.24 SYSTEM2

Statement: SCS

Modes :

Syntax : SCS Address or Register

See SCC

1.25 SYSTEM2

Statement: SEQ

Modes :

Syntax : SEQ Address or Register

See SCC

1.26 SYSTEM2

Statement: SGE

Modes :

Syntax : SGE Address or Register

See SCC

1.27 SYSTEM2

Statement: SGT

Modes :

Syntax : SGT Address or Register

See SCC

1.28 SYSTEM2

Statement: SHI

Modes :

Syntax : SHI Address or Register

See SCC

1.29 SYSTEM2

Statement: SLE

Modes :

Syntax : SLE Address or Register

See SCC

1.30 SYSTEM2

Statement: SLS

Modes :

Syntax : SLS Address or Register

See SCC

1.31 SYSTEM2

Statement: SLT

Modes :

Syntax : SLT Address or Register

See SCC

1.32 SYSTEM2

Statement: SMI

Modes :
Syntax : SMI Address or Register

See SCC

1.33 SYSTEM2

Statement: SNE

Modes :
Syntax : SNE Address or Register

See SCC

1.34 SYSTEM2

Statement: SPL

Modes :
Syntax : SPL Address or Register

See SCC

1.35 SYSTEM2

Statement: SVC

Modes :
Syntax : SVC Address or Register

See SCC

1.36 SYSTEM2

Statement: SVS

Modes :
Syntax : SVS Address or Register

See SCC

1.37 SYSTEM2

Statement: ST

Modes :
Syntax : ST

Stop work. (careful!) Leads to a TRAPV Exception.

1.38 SYSTEM2

Statement: SF

Modes :
Syntax : SF

1.39 SYSTEM2

Statement: Stop

Modes :
Syntax : Stop

The Stop command will cause program flow to stop, and user control to be transferred to Blitz 2 direct mode.

The Stop command is really only useful in debugging situations, as it allows the programmer a chance to have a look at program variables via Blitz 2's direct mode.

1.40 SYSTEM2

Statement: SUB

Modes :
Syntax : SUB Source operand, Destination operand

Binary subtraction.

1.41 SYSTEM2

Statement: SUBI

Modes :
Syntax : SUBI Direct value,Address or Register

Subtract a constant.

1.42 SYSTEM2

Statement: SUBA

Modes :
Syntax : SUBA Address or Register,A0-6

Binary subtraction from an address register

1.43 SYSTEM2

Statement: SUBQ

Modes :
Syntax : SUBQ Direct value,Address or Register

Fast subtraction of a three bit constant.

1.44 SYSTEM2

Statement: SUBX

Modes :
Syntax : SUBX Source operand,Destination operand

Subtraction with transfer to X flag

1.45 SYSTEM2

Statement: SWAP

Modes :
Syntax : SWAP D0-7

Exchange the two halves of the register (the upper and lower 16 bits)

1.46 SYSTEM2

Statement: TAS

Modes :

Syntax : TAS Direct value

Test a bit and set bit 7

1.47 SYSTEM2

Statement: TRAP

Modes :

Syntax : TRAP Direct value

Jump to an Exception

1.48 SYSTEM2

Statement: TRAPV

Modes :

Syntax : TRAPV

Check if overflow flag set, then TST Address or Register.
Test an operand and set the N and Z flag

1.49 SYSTEM2

Statement: TST

Modes :

Syntax : TST

1.50 SYSTEM2

Statement: UNLK

Modes :

Syntax : UNLK A0-6

Un-link stack area

1.51 SYSTEM2

Statement: nana

Modes :
Syntax : nana

1.52 SYSTEM2

Statement: List

Modes :
Syntax : List

1.53 SYSTEM2

Statement: SetErr

Modes :
Syntax : SetErr

The SetErr command allows you to set up custom error handlers. Program code which appears after the SetErr command will be executed when any Blitz 2 runtime errors are caused. Custom error code should be ended by an End SetErr.

1.54 SYSTEM2

Statement: ClrErr

Modes :
Syntax : ClrErr

ClrErr may be used to remove a custom error handler set up using SetErr.

1.55 SYSTEM2

Statement: Errn

Modes :
Syntax : Errn

1.56 SYSTEM2

Statement: ErrFail

Modes :
Syntax : ErrFail

ErrFail may be used within custom error handlers to cause a 'normal' error. The error which caused the custom error handler to be executed will reported and transfer will be passed to direct mode.

1.57 SYSTEM2

Statement: Addr

Modes :
Syntax : Addr Objectname(Object#)

Addr is a low-level function allowing advanced programmers the ability to find where a particular Blitz 2 object resides in RAM. An appendix at the end of this manual lists all Blitz 2 object formats.

1.58 SYSTEM2

Statement: MOD

Modes :
Syntax : MOD

1.59 SYSTEM2

Statement: Pi

Modes :
Syntax : Pi

1.60 SYSTEM2

Statement: Repeat

Modes :

Syntax : Repeat

Repeat is used to begin a Repeat...Until loop. Each Repeat statement in a program must have a corresponding Until further down the program.

The purpose of Repeat/Until loops is to cause a section of code to be executed AT LEAST ONCE before a test is made to see if the code should be executed again.

1.61 SYSTEM2

Statement: Until

Modes :

Syntax : Until Expression

Until is used to terminate a Repeat/Until loop. If Expression proves to be true (non 0), then program flow will continue from the command following Until. If Expression proves to be false (0), then program flow will go back to the corresponding Repeat, found further up the program.

1.62 SYSTEM2

Statement: PutReg

Modes :

Syntax : PutReg 68000 Reg,Variable

PutReg may be used to transfer a value from any 68000 register (d0-d7/a0-a7) into a BASIC variable. If the specified variable is a string, long, float or quick, then all 4 bytes from the register will be transferred. If the specified variable is a word or a byte, then only the relevant low bytes will be transferred.

1.63 SYSTEM2

Statement: Pop

Modes :

Syntax : Pop Gosub|For|Select|If|While|Repeat

Sometimes, it may be necessary to exit from a particular type of program loop in order to transfer program flow to a different part of the program. However, to achieve this Blitz 2 must be told that the relevant loop should be 'forgotten'. This is the purpose of Pop.

Actually, Pop is only necessary to prematurely terminate Gosubs, Fors and Selects. If, While and Repeat have been included for completeness.

1.64 SYSTEM2

Statement: INCDIR

Modes :

Syntax : INCDIR Pathname

The INCDIR command allows you to specify an AmigaDos path to be prefixed to any filenames specified by any of INCLUDE, XINCLUDE or INCBIN commands.

Pathname may be optionally quote enclosed to avoid tokenisation problems.

1.65 SYSTEM2

Statement: EndIf

Modes :

Syntax : EndIf

EndIf is used to terminate an 'If block'. An If block is begun by use of the If statement. Please refer to If for more information on If blocks.

1.66 SYSTEM2

Statement: Wend

Modes :

Syntax : Wend

Wend is used in conjunction with While to determine a section of program to be executed repeatedly based upon the truth of an expression.

1.67 SYSTEM2

Statement: SysJsr

Modes :

Syntax : SysJsr Routine

SysJsr allows you to call any of Blitz 2's system routines from your own program. Routine specifies a routine number to call.

1.68 SYSTEM2

Statement: WBStartup

Modes :

Syntax : WBStartup

By executing WBStartup at some point in your program, your program will be given the ability to run from Workbench. A program run from Workbench which does NOT include the WBStartup command will promptly crash if an attempt is made to run it from Workbench.

1.69 SYSTEM2

Statement: Maximum

Modes :

Syntax : Maximum Objectname

The Maximum function allows a program to determine the 'maximum' setting for a particular Blitz 2 object. Maximum settings are entered into the 'OPTIONS' requester, accessed through the 'COMPILER' menu of the Blitz 2 editor.

1.70 SYSTEM2

Statement: On

Modes :

Syntax : On Expression Goto|Gosub Program Label[,Program Label...]

1.71 SYSTEM2

Statement: Off

Modes :
Syntax : Off

1.72 SYSTEM2

Statement: Forever

Modes :
Syntax : Forever

Forever may be used instead of Until to cause a Repeat/Until loop to NEVER exit.

Executing Forever is identical to executing 'Until 0'.

1.73 SYSTEM2

Statement: Restore

Modes :
Syntax : Restore [Program Label]

Restore allows you to set Blitz 2's internal 'data pointer' to a particular piece of data. after executing a Restore, The first item of data following the specified Program Label will become the data to be read when the next Read command is executed.

Restore with no parameters will reset the data pointer to the very first piece of data in the program.

1.74 SYSTEM2

Statement: Exchange

Modes :
Syntax : Exchange Var,Var

Exchange will 'swap' the values contained in the 2 specified variables. Exchange may only be used with 2 variables of the same type.

1.75 SYSTEM2

Statement: USEPATH

Modes :

Syntax : USEPATH Pathtext

```
aliens()\x=160
aliens()\y=100
aliens()\xs=10
aliens()\ys=-10
```

USEPATH can be used to save you some typing, like so:

```
USEPATH aliens()
\x=160
\y=100
\xs=10
\ys=-10
```

Whenever Blitz2 encounters a variable starting with the backslash character ('\''), it simply inserts the current USEPATH text before the backslash.

1.76 SYSTEM2

Statement: CloseEd

Modes :

Syntax : CloseEd

The CloseEd statement will cause the Blitz 2 editor screen to 'close down' when programs are executed from within Blitz 2. This may be useful when writing programs which use a large amount of chip memory, as the editor screen itself occupies about 40K of chip memory.

CloseEd will have no effect on executable files run outside of the Blitz 2 environment.

1.77 SYSTEM2

Statement: NoCli

Modes :

Syntax : NoCli

NoCli will prevent the normal 'Default Cli' from opening when programs are executed from within Blitz 2. NoCli has no effect on executable files run outside of the Blitz 2 environment.

1.78 SYSTEM2

Statement: BitTst

Modes :
Syntax : BitTst

1.79 SYSTEM2

Statement: BitSet

Modes :
Syntax : BitSet

1.80 SYSTEM2

Statement: BitClr

Modes :
Syntax : BitClr

1.81 SYSTEM2

Statement: BitChg

Modes :
Syntax : BitChg

1.82 SYSTEM2

Statement: Used

Modes :
Syntax : Used Objectname (Object#)

Used returns the currently used object number. This is useful for routines which need to operate on the currently used object, also interrupts should restore currently used object settings.

1.83 ;name

Statement: Runerrson ;name

Modes :

Syntax : Runerrson ;name

1.84 SYSTEM2

Statement: Runerrsoff

Modes :

Syntax : Runerrsoff

See description of Runerrson.

1.85 SYSTEM2

SYSTEM2

Overview

Command Index

MOVEM

MOVEQ

MULS

MULU

NBCD

NEG

NEGX

NOP

NOT

OR

ORI

PEA

rese

ROL

ROR

ROXL

ROXR

RTE

RTR

RTS

Runerrsoff

SBCD

SCC

SCS

SEQ

SF

SGE

SGT

SHI

SLE

SLS

SLT

SMI

SNE

SPL

ST

Stop

SUB

SUBA

SUBI

SUBQ

SUBX

SVC

SVS

SWAP

TAS

TRAP

TRAPV

TST

UNLK
