Memory Commands

Instructions and functions to handle memory access

Poke ADDRESS, VALUE
Change a one-byte word at a memory address
Parameters:
ADDRESS: The address to change, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.
VALUE: The value to set

Doke ADDRESS, VALUE
Change a two-byte word at a memory address. Value is set in little-endian in PC mode and big-endian in Amiga mode
Parameters:
ADDRESS: The address to change, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.
VALUE: The value to set

Loke ADDRESS, VALUE
Change a four-byte word at a memory address. Value is set in little-endian in PC mode and big-endian in Amiga mode
Parameters:
ADDRESS: The address to change, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.
VALUE: The value to set

Poke$ ADDRESS, TEXT$
Write the ascii values of a string in memory
Parameters:
ADDRESS: The address to write to, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.
TEXT$: The string to write

Peek$ ADDRESS, LENGTH, STOP
Read a string from memory
Parameters:
ADDRESS: The address to read, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.
LENGTH: The number of bytes to read.
STOP: The ascii code of the character signifying the end of the string. Default is 0
Value returned:
string: The string contained at the address

Peek ADDRESS
read a byte from an address
Parameters:
ADDRESS: The address to change, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.
Value returned:
integer: The value contained at the address

Deek ADDRESS

Read a two-bytes value from an address. Value is read in little-endian in PC mode and big-endian in Amiga mode

Parameters:

ADDRESS: The address to change, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.

Value returned:

integer: The value contained at the address

Leek ADDRESS

Read a four-bytes value from an address. Value is read in little-endian in PC mode and big-endian in Amiga mode

Parameters:

ADDRESS: The address to change, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.

Value returned:

integer: The value contained at the address

Free

Return the amount of free memory on the machine. Warnign, not tested and maybe deprecated in the future./ The original AMOS instruction reported the amount of free memory in the variable buffer area

Value returned:

integer: The amount of free memory

Chip Free

Return the amount of free memory on the machine. Warnign, not tested and maybe deprecated in the future./ The original AMOS instruction reported the amount of free chip memory of the machine, which has no meaning tocay

Value returned:

integer: The amount of free memory

Fast Free

Return the amount of free memory on the machine. Warnign, not tested and maybe deprecated in the future./ The original AMOS instruction reported the amount of free fast memory of the machine, which has no meaning tocay

Value returned:

integer: The amount of free memory

Fill ... To ... START, FINISH, PATTERN

fill memory block with the contents of a variable.

Parameters:

START: The address to change, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.

FINISH: The address of the end of the area to fill

PATTERN: The four-byte pattern that will be repeated when filling

Copy ... To ... START, FINISH, DESTINATION

Copy a memory block

Parameters:

START: The address of the block to copy, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.

FINISH: The address of the end of the area to copy

DESTINATION: The destination address, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.

Hunt ... To ... START, FINISH, TEXT$
Find a string of characters in memory
Parameters:
START: The address of the block to copy, should be resolved in AOZ by using the "Start" function to get the address of a bank (magical number) and adding the offset to the address. Normal computer memory is inaccessible.
FINISH: The address of the end of the area to copy
TEXT$: The string to look for
Value returned:
integer: The position of the first character if found, zero if not found

BSet VARIABLE, POSITION
Set a bit of a variable to 1
Parameters:
VARIABLE: The integer variable to change
POSITION: The number of the bit to set

BClr VARIABLE, POSITION
Set a bit of a variable to 0
Parameters:
VARIABLE: The integer variable to change
POSITION: The number of the bit to clear

BChg VARIABLE, POSITION
Toggle a bit of a variable
Parameters:
VARIABLE: The integer variable to change
POSITION: The number of the bit to toggle

BTst POSITION, VALUE
Test a bit in an interger number
Parameters:
POSITION: The number of the bit to test
VALUE: The number to test
Value returned:
integer: True if the bit is set to one, False if not

Ror.b NUMBER, VARIABLE
Rotate the content of an integer variable to the right on the first 8 bits
Parameters:
NUMBER: The number or shifts
VARIABLE: The integer variable to rotate

Ror.w NUMBER, VARIABLE
Rotate the content of an integer variable to the right on the first 16 bits
Parameters:
NUMBER: The number or shifts
VARIABLE: The integer variable to rotate

Ror.l NUMBER, VARIABLE
Rotate the content of an integer variable to the right on 32 bits

Parameters:
NUMBER: The number or shifts
VARIABLE: The integer variable to rotate

Rol.b NUMBER, VARIABLE
Rotate the content of an integer variable to the left on the first 8 bits
Parameters:
NUMBER: The number or shifts
VARIABLE: The integer variable to rotate

Rol.w NUMBER, VARIABLE
Rotate the content of an integer variable to the left on the first 16 bits
Parameters:
NUMBER: The number or shifts
VARIABLE: The integer variable to rotate

Rol.l NUMBER, VARIABLE
Rotate the content of an integer variable to the left on 32 bits
Parameters:
NUMBER: The number or shifts
VARIABLE: The integer variable to rotate

Exec COMMAND$, OUTPUT$
Execute a system command. This instruction will be implemented in executable and node.js transpiled applications
Parameters:
COMMAND$: The command with it parameters
OUTPUT$: The DOS-like output, like "StdOut"

Set Stack SPACE
Deprecated, wa used to set the stack space on the Amiga
Parameters:
SPACE: The length opf the stack

Set Equate Bank BANK_NUMBER
Deprecated, was used on the Amiga
Parameters:
BANK_NUMBER:

Call ADDRESS, PARAMETER
Deprecated, was used on the Amiga
Parameters:
ADDRESS:
PARAMETER:

ExeCall OFFSET
Deprecated, was used on the Amiga
Parameters:
OFFSET:

GfxCall OFFSET
Deprecated, was used on the Amiga

Parameters:

OFFSET:

DosCall OFFSET

Deprecated, was used on the Amiga

Parameters:

OFFSET:

IntCall OFFSET

Deprecated, was used on the Amiga

Parameters:

OFFSET: