Komanetsky Functions for Linux
Version: 2023/03/31

Bill Komanetsky
Las Positas College

March 31, 2023

Any printed or copied version of this document are for reference only and should not be considered the most up to date version of this document. Please see the version of this document located on the Canvas class management system for the most recent version

Contents

1 Calling Functions

The Komanetsky Linux functions use the push-arguments format of calling functions. This means that any function will receive its arguments sent to it via the stack.

Example of such a call:

```
push argument-1
push [argument...]
push [argument-n]
call functionname
```

Any return value will come via the eax or rax registers (32-bit or 64-bit respectively)

2 Operating System Compatibility

The Komanetsky functions for Linux is compatible with both 32-bit and 64-bit Linux applications. Each function in the list of functions within this document notes the differences the function(s) may have across operating systems (32 or 64 bit).

2.1 32-bit Libraries

The following files need to be included in your assembly/link steps

- functions.inc: The include file to go into any assembly source code file that wishes to use the functions
- functions.o: The object file which must be linked with any object file using the functions

2.2 64-bit Libraries

The following files need to be included in your assembly/link steps

- functions64.inc: The include file to go into any assembly source code file that wishes to use the functions
- functions64.o: The object file which must be linked with any object file using the functions

3 Listing of Functions

3.1 Printing Text to StdOut

3.1.1 PrintCenter

Description The PrintCenter function is used for printing a null terminated string to stdout, center justified within a set width

Arguments

- The Address of the string to print
- The width of the column to center justify the text within

Return Value(s)

- None

Example

```
.data
   .stringLabel db ”Hello There”, 0h
.code
   push stringLabel
   push QWORD 80
   call PrintCenter
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed

3.1.2 PrintLeft

Description The PrintLeft function is used for printing a null terminated string to stdout, left justified within a set width

Arguments

- The Address of the string to print
- The width of the column to left justify the text within

Return Value(s)

- None

Example

```
.data
    .stringLabel db ”Hello There”, 0h
.code
    push stringLabel
    push QWORD 80
    call PrintLeft
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed

---

3.1.3 PrintString

Description The PrintString function is used for printing text to stdout based on the address of the string only. Your string should be null terminated to assure the printing of the string ends at the appropriate location

Arguments

- The Address of the string to print

Return Value(s)

- None

Example

```
.data
    .stringLabel db ”Hello There”, 0h
.code
    push stringLabel
    call PrintString
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed

---

3.1.4 PrintText

Description The PrintText function is used for printing text to stdout based on the address of the string and its length.

Arguments

- The Address of the string to print
- The length of the string to print

Return Value(s)

- None

Example

```
.data
    .stringLabel db ”Hello There”
.code
    push stringLabel
    push 11
    call PrintText
```

Usage Notes

- This function does not recognize a null terminated string. It relys on the length of the string specified as the second argument pushed into the stack
- No carriage return or line feed will be printed after the string or sub-string is printed

---

3.1.5 PrintRight

Description The PrintRight function is used for printing a null terminated string to stdout, right justified within a set width

Arguments

- The Address of the string to print
- The width of the column to right justify the text within

Return Value(s)

- None

Example

```
.data
    .stringLabel db ”Hello There”, 0h
.code
    push stringLabel
    push QWORD 80 ;32 bit: push DWORD 80
    call PrintRight
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed

---

3.1.6 PrintStackFrame

Description The PrintStackFrame function is used for printing the contents of a function's stack frame to stdout.

Arguments

- None

Return Value(s)

- None

Example

```
.data
.code
    call PrintStackFrame
```

Usage Notes

- No carriage return or line feed will be printed after the stack frame's contents is printed
- This function assumes you have created a valid stack frame
- Call this function after the stack frame is created and after all local function variables have been allocated and before you use the stack for any other purpose

---

3.1.7 PrintComma

Description The PrintComma function is used for printing a single comma character to stdout

Arguments

- None

Return Value(s)

- None

Example

```
.data
.code
    call PrintComma
```

Usage Notes

- No carriage return or line feed will be printed after the stack frame's contents is printed

---

3.1.8 Printendl

Description The Printendl function is used for printing a new-line and carriage return characters to stdout

Arguments

- None

Return Value(s)

- None

Example

```
.data
.code
    call Printendl
```

Usage Notes

- No carriage return or line feed will be printed after the stack frame's contents is printed

---

3.1.9 PrintSpace

Description The PrintSpace function is used for printing a single space character to stdout

Arguments

- None

Return Value(s)

- None

Example

```
.data
.code
    call PrintSpace
```

Usage Notes

- No carriage return or line feed will be printed after the stack frame's contents is printed

---

3.1.10 PrintSpaces

Description The PrintSpaces function is used for printing 'n' number of space characters to stdout

Arguments

- The number of spaces to print

Return Value(s)

- None

Example

```
.data
.code
    push QWORD 5 ;32 bit: push DWORD 5
    call PrintSpace
```

Usage Notes

- The above example will print 5 spaces because of the double word 5 pushed onto the stack
- No carriage return or line feed will be printed after the stack frame's contents is printed

---

3.1.11 PrintTab

Description The PrintTab function is used for printing a tab character to stdout

Arguments

- None

Return Value(s)

- None

Example

```
.data
.code
    call PrintTab
```

Usage Notes

- No carriage return or line feed will be printed after the stack frame's contents is printed

3.2 Printing Numeric values to StdOut

3.2.1 Print64bitNumBinary/Print32bitNumBinary

Description The Print64bitNumBinary function is used for printing a 64-bit (8 byte) number to stdout.in binary format

Arguments

- The number to print

Return Value(s)

- None

Example

```
.data
    someValue dq 12345h ;32 bit: someValue dd 12345h
.code
    push QWORD [someValue] ;32 bit: push DWORD [someValue]
    call Print64bitNumBinary
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed
- The letter 'b' will be printed after the hexidecimal number
- Significant zero bits will not be printed
- Print64bitNumBinary is available only in the 64-bit functions libraries
- Print32bitNumBinary is available only in the 32-bit functions libraries

3.2.2 Print64bitNumHex/Print32bitNumHex

Description The Print64bitNumHex function is used for printing a 64-bit (8 byte) number to stdout.in hexadecimal format

Arguments

- The number to print

Return Value(s)

- None

Example

```
.data
    someValue dq 12345h ; 32 bit: someValue dd 12345h
.code
    push QWORD [someValue] ; 32 bit: push DWORD [someValue]
    call Print64bitNumHex
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed
- The letter 'h' will be printed after the hexidecimal number
- Print64bitNumHex is available only in the 64-bit functions libraries
- Print32bitNumHex is available only in the 32-bit functions libraries

---

3.2.3 Print64bitSNumDecimal/Print32bitNumSDecimal

Description The Print64bitSNumDecimal function is used for printing a 64-bit (8 byte) signed number to stdout.in decimal format

Arguments

- The signed number to print

Return Value(s)

- None

Example

```
.data
    someValue dq -12345 ; 32 bit: someValue dd -12345
.code
    push QWORD [someValue] ; 32 bit: push DWORD [someValue]
    call Print64bitSNumDecimal
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed
- Print64bitSNumDecimal is available only in the 64-bit functions libraries
- Print32bitSNumDecimal is available only in the 32-bit functions libraries

---

3.2.4 Print64bitNumDecimal/Print32bitNumDecimal

Description The Print64bitNumDecimal function is used for printing a 64-bit (8 byte) number to stdout.in decimal format

Arguments

- The number to print

Return Value(s)

- None

Example

```
.data
    someValue dq 12345 ; 32 bit: someValue dd 12345
.code
    push QWORD [someValue] ; 32 bit: push DWORD [someValue]
    call Print64bitNumDecimal
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed
- Print64bitNumDecimal is available only in the 64-bit functions libraries
- Print32bitNumDecimal is available only in the 32-bit functions libraries

---

3.2.5 Print64bitNumOctal/Print32bitNumOctal

Description The Print64bitNumOctal function is used for printing a 64-bit (8 byte) number to stdout.in Octal format

Arguments

- The number to print

Return Value(s)

- None

Example

```
.data
    someValue dq 12345h ; 32 bit: someValue dd 12345h
.code
    push QWORD [someValue] ; 32 bit: push DWORD [someValue]
    call Print64bitNumOctal
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed
- The letter 'o' will be printed after the octal number
- Print64bitNumOctal is available only in the 64-bit functions libraries
- Print32bitNumOctal is available only in the 32-bit functions libraries

---

3.2.6 PrintQWFloat/PrintDWFloat

Description The PrintQWFloat function is used for printing a 64-bit (8 byte) floating point number to stdout in decimal format. PrintDWFloat is used for printing 32-bit (4 byte) floating point number to stdout in decimal format

Arguments

- The number to print
- The number of decimal points to print

Return Value(s)

- None

Example

```
.data
    someValue dq 123.45 ; 32 bit: someValue dd 123.45
.code
    push QWORD [someValue] ; 32 bit: push DWORD [someValue]
    push QWORD 4 ; 32 bit: push DWORD 4
    call PrintQWFloat
```

Usage Notes

- The above example will print the floating point number with 4 digits of fractional accuracy
- No carriage return or line feed will be printed after the string or sub-string is printed
- PrintQWFloat is available only in the 64-bit functions libraries
- PrintDWFloat is available only in the 32-bit functions libraries

---

3.2.7 Print64bitFloatBinary/Print32bitFloatBinary

Description The Print64bitFloatBinary function is used for printing a 64-bit (8 byte) floating point number to stdout.in binary format

Arguments

- None

Return Value(s)

- None

Example

```
.data
    someValue dq 123.45 ; 32 bit: someValue dd 123.45
.code
    push QWORD [someValue] ; 32 bit: push DWORD [someValue]
    call Print64bitFloatBinary
```

Usage Notes

- The above example will print the floating point number with 4 digits of fractional accuracy
- No carriage return or line feed will be printed after the string or sub-string is printed
- Print64bitFloatBinary is available only in the 64-bit functions libraries
- Print32bitFloatBinary is available only in the 32-bit functions libraries

---

3.2.8 IsNAN

Description The IsNAN function is used for to indicate if a floating point number is in an invalid format

Arguments

- None

Return Value(s)

- The carry flag will be set if the floating point number is a +NAN, -NAN, +Infinity or -Infinity value

Example

```
.data
   someValue dq 123.45 ; 32 bit: someValue dd 123.45
.code
   push QWORD [someValue] ; 32 bit: push DWORD [someValue]
   call IsNAN
```

Usage Notes

- The above example will print the floating point number with 4 digits of fractional accuracy
- No carriage return or line feed will be printed after the string or sub-string is printed

---

3.2.9 PrintQWFloatSN/PrintDWFloatSN

Description The PrintQWFloatSN function is used for printing a 64-bit (8 byte) floating point number to stdout in Decimal/Scientific Notation format. PrintDWFloatSN function is used for printing a 32-bit (4 byte) floating point number to stdout in Decimal/Scientific Notation format

Arguments

- The number to print
- The number of decimal points to print

Return Value(s)

- None

Example

```
.data
    someValue dq 123.45 ; 32 bit: someValue dd 123.45
.code
    push QWORD [someValue] ; 32 bit: push DWORD [someValue]
    push QWORD 4 ; 32 bit: push DWORD 4
    call PrintQWFloatSN
```

Usage Notes

- The above example will print the floating point number with 4 digits of fractional accuracy
- No carriage return or line feed will be printed after the string or sub-string is printed
- PrintQWFloatSN is available only in the 64-bit functions libraries
- PrintDWFloatSN is available only in the 32-bit functions libraries

---

3.2.10 PrintRegisters

Description The PrintRegisters function is used for printing all CPU registers to StdOut

Arguments

- None

Return Value(s)

- None

Example

```
.data
.bss
.code
    call PrintRegisters
```

Usage Notes

- No carriage return or line feed will be printed after the registers are printed

---

3.2.11 PrintFloatingRegisters

Description The PrintFloatingRegisters function is used for printing all floating point registers to StdOut

Arguments

- None

Return Value(s)

- None

Example

```
.data
.bss
.code
    call PrintFloatingRegisters
```

Usage Notes

- No carriage return or line feed will be printed after the registers are printed

---

3.2.12 PrintFlags

Description The PrintFlags function is used for printing the most common EFLAGs' status to StdOut. If a flag is set, its status will be printed. If a flag is not set, nothing will be printed for that particular flag. The flags which will be checked include:

- Carry Flag
- Overflow Flag
- Zero Flag
- Sign Flag
- Parity Flag

Arguments

- None

Return Value(s)

- None

Example

```
.data
.bss
.code
    call PrintFlags
```

Usage Notes

- None

3.3 Printing Numeric Arrays to StdOut

3.3.1 PrintByteArray

Description The PrintByteArray function is used for printing the contents of a byte array to stdout in hexidecimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    bArray db 10h, 20h, 30h
      .LENGTHOF equ ($-bArray)
.code
    push bArray
    push bArray.LENGTHOF
    call PrintByteArray
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed

3.3.2 PrintByteArrayDec

Description The PrintByteArrayDec function is used for printing the contents of a byte array to stdout in decimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    bArray db 10, 20, 240
      .LENGTHOF equ ($-bArray)
.code
    push bArray
    push bArray.LENGTHOF
    call PrintByteArrayDec
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed

---

3.3.3 PrintSByteArrayDec

Description The PrintSByteArrayDec function is used for printing the contents of a byte array to stdout in signed decimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    bArray db -10, 20, -240
      .LENGTHOF equ ($-bArray)
.code
    push bArray
    push bArray.LENGTHOF
    call PrintSByteArrayDec
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed

---

3.3.4 PrintWordArray

Description The PrintWordArray function is used for printing the contents of a word array to stdout in hexidecimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    wArray dw 10h, 20h, 30h
      .LENGTHOF equ ($-wArray)/2
.code
    push wArray
    push wArray.LENGTHOF
    call PrintWordArray
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed

---

3.3.5 PrintWordArrayDec

Description The PrintWordArrayDec function is used for printing the contents of a word array to stdout in decimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    wArray dw 10, 20, 240
      .LENGTHOF equ ($-wArray)/2
.code
    push wArray
    push wArray.LENGTHOF
    call PrintWordArrayDec
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed

---

3.3.6 PrintWordArrayDec

Description The PrintWordArrayDec function is used for printing the contents of a word array to stdout in signed decimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    wArray dw -10, 20, -240
      .LENGTHOF equ ($-wArray)/2
.code
    push wArray
    push wArray.LENGTHOF
    call PrintWordArrayDec
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed

---

3.3.7 PrintDWordArray

Description The PrintDWordArray function is used for printing the contents of a dobule word array to stdout in hexidecimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    dArray dd 10h, 20h, 30h
      .LENGTHOF equ ($-dArray)/4
.code
    push dArray
    push dArray.LENGTHOF
    call PrintDWordArray
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed

---

3.3.8 PrintDWordArrayDec

Description The PrintDWordArrayDec function is used for printing the contents of a double word array to stdout in decimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    dArray dd 10, 20, 240
      .LENGTHOF equ ($-dArray)/4
.code
    push dArray
    push dArray.LENGTHOF
    call PrintDWordArrayDec
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed

---

3.3.9 PrintSDWordArrayDec

Description The PrintSDWordArrayDec function is used for printing the contents of a double word array to stdout in signed decimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    dArray dd 10, 20, 240
      .LENGTHOF equ ($-dArray)/4
.code
    push dArray
    push dArray.LENGTHOF
    call PrintSDWordArrayDec
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed

---

3.3.10 PrintQWordArray

Description The PrintQWordArray function is used for printing the contents of a Quad-Word array to stdout in hexadecimal format.

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    qArray dq 100h, 200h, 300h
      .LENGTHOF equ ($-qArray)/8
.code
    push qArray
    push qArray.LENGTHOF
    call PrintQWordArray
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed
- This function is available only in the 64-bit functions libraries

---

3.3.11 PrintQWordArrayDec

Description The PrintQWordArrayDec function is used for printing the contents of a Quad-Word array to stdout in decimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    qArray dq 100h, 200h, 300h
      .LENGTHOF equ ($-qArray)/8
.code
    push qArray
    push qArray.LENGTHOF
    call PrintQWordArrayDec
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed
- This function is available only in the 64-bit functions libraries

---

3.3.12 PrintSQWordArrayDec

Description The PrintSQWordArrayDec function is used for printing the contents of a Quad-Word array to stdout in signed decimal format

Arguments

- Address of the array to print
- The number of entries in the array

Return Value(s)

- None

Example

```
.data
    qArray dq -10, 256, -1786
      .LENGTHOF equ ($-qArray)/8
.code
    push qArray
    push qArray.LENGTHOF
    call PrintSQWordArrayDec
```

Usage Notes

- No carriage return or line feed will be printed after the array is printed
- This function is available only in the 64-bit functions libraries

3.4 Input from StdIn

3.4.1 ClearKBuffer

Description The ClearKBuffer function is used to clear the StdIn keyboard buffer. At times, numeric input may not be possible because of the existence of characters in the operating system's StdIn buffer. This function may be effective at clearing that buffer before numeric input is attempted

Arguments

- None

Return Value(s)

- None

Example

```
.data
.bss
.code
    call ClearKBuffer
```

Usage Notes

- this function may only be effective before imputing numeric values using any of the documented numeric input functions which are part of this document

3.4.2 ReadText

Description The ReadText function is used for inputing data from StdIn into a programmer defined memory location

Arguments

- Address of the input buffer
- The size (in bytes) of the input buffer

Return Value(s)

- The RAX/EAX register will contain the total characters read from StdIn

Example

```
.data
.bss
    readBuffer resb 255
      .LENGTHOF equ ($-readBuffer)
.code
    push readBuffer
    push readBufferLENGTHOF
    call ReadText
```

Usage Notes

- RAX/EAX will contain the total characters entered by the user (from stdin)
- The data input will be stored in the address specified by the first argument placed in the stack before the function is called

---

3.4.3 ReadUInt

Description The ReadUInt function is used for imputing an unsigned integer value from StdIn

Arguments

- None

Return Value(s)

- The RAX/EAX register will contain the integer read from StdIn
- The carry flag will be set if the number entered is invalid (contains spaces, decimal points, non-numeric characters)

Example

```
.data
.bss
.code
    call ReadUInt
    jnc validIntegerValue
    ;take action on invalid integer input
```

Usage Notes

- RAX/EAX will contain the integer read from StdIn

3.4.4 ReadSInt

Description The ReadUInt function is used for imputing a signed integer value from StdIn

Arguments

- None

Return Value(s)

- The RAX/EAX register will contain the signed integer read from StdIn
- The carry flag will be set if the number entered is invalid (contains spaces, decimal points, non-numeric characters)

Example

```
.data
.bss
.code
  call ReadSInt
  jnc validIntegerValue
  ;take action on invalid integer input
```

Usage Notes

- RAX/EAX will contain the integer read from StdIn

3.4.5 ReadTextWPrompt

Description The ReadTextWPrompt function is used for inputting data from StdIn into a programmer defined memory location after the display of a null terminated string sent to StdIn

Arguments

- Address of prompt string
- Address of the input buffer
- The size (in bytes) of the input buffer

Return Value(s)

- The RAX/EAX register will contain the total characters read from StdIn

Example

```
.data
  promptString db ”Please enter any string: ”, 0h
.bss
  readBuffer resb 255
   .LENGTHOF equ ($-readBuffer)
.code
  push promptString
  push readBuffer
  push readBufferLENGTHOF
  call ReadTextWPrompt
```

Usage Notes

- RAX/EAX will contain the total characters entered by the user (from stdin)
- The data input will be stored in the address specified by the first argument placed in the stack before the function is called
- This function executes a PrintString and then a InputText function in a single function call

---

3.4.6 InputFloat

Description The InputFloat function is used for inputting a decimal formatted floating point number from StdIn

Arguments

- None

Return Value(s)

- The RAX/EAX and ST(0) registers will contain the floating point number read from StdIn in IEEE 754 format
- The carry flag will be set if the number entered is invalid (contains spaces, decimal points, non-numeric characters)

Example

```
.data
.bss
.code
    call InputFloat
    jnc validFloatValue
    ;take action on invalid float input
```

Usage Notes

- RAX/EAX will contain the floating point number read from StdIn in IEEE 754 format

---

3.4.7 InputBin

Description The InputBin function is used for inputting a binary number from StdIn

Arguments

- None

Return Value(s)

- The RAX/EAX register will contain the binary number read from StdIn
- The carry flag will be set if the number entered is invalid (contains spaces, decimal points, non-numeric characters, numeric characters other than 1's or 0's)

Example

```
.data
.bss
.code
    call InputBin
    jnc validBinValue
    ;take action on invalid Binary input
```

Usage Notes

- RAX/EAX will contain the binary number read from StdIn

3.4.8 InputHex

Description The InputHex function is used for inputting a hexadecimal number from StdIn

Arguments

- None

Return Value(s)

- The RAX/EAX register will contain the hexadecimal number read from StdIn
- The carry flag will be set if the number entered is invalid (contains spaces, decimal points, non-numeric characters, non-hexadecimal characters)

Example

```
.data
.bss
.code
    call InputHex
    jnc validHexValue
    ;take action on invalid hexadecimal input
```

Usage Notes

- RAX/EAX will contain the hexadecimal number read from StdIn

3.5 String Manipulation Functions

3.5.1 ToUpper

Description Convert all lower case ASCII alphabetic characters to upper case

Arguments

- Address of the array of bytes to process

Return Value(s)

- RAX/EAX will contain the number of bytes converted

Example

```
.data
    myString db ”This is My String 123 to Convert”, 0h
.bss
.code
    push mySTring
    call ToUpper
```

Usage Notes

- The array of characters must be zero/null terminated
- Any non-alphabetic ascii value will not be converted

3.5.2 ToLower

Description Convert all upper case ASCII alphabetic characters to lower case

Arguments

- Address of the array of bytes to process

Return Value(s)

- RAX/EAX will contain the number of bytes converted

Example

```
.data
    myString db ”This is My String 123 to Convert”, 0h
.bss
.code
    push mySTring
    call ToLower
```

Usage Notes

- The array of characters must be zero/null terminated
- Any non-alphabetic ascii value will not be converted

3.5.3 AtoUInt

Description Convert an ascii numeric unsinged integer string into the RAX/EAX register

Arguments

- Address of the unsigned integer string

Return Value(s)

- RAX/EAX will contain the unsigned binary value

Example

```
.data
   myString db ”1234567”, 0h
.bss
.code
   push myString
   call AtoUInt
```

Usage Notes

- The ascii string must be zero/null terminated
- The carry flag will be set if an invalid ascii number is provided

---

3.5.4 AtoSInt

Description Convert an ascii numeric signed integer string into the RAX/EAX register

Arguments

- Address of the signed integer string

Return Value(s)

- RAX/EAX will contain the unsigned binary value

Example

```
.data
   myString db ”-1234567”, 0h
.bss
.code
   push myString
   call AtoSInt
```

Usage Notes

- The ascii string must be zero/null terminated
- The carry flag will be set if an invalid ascii number is provided

---

3.5.5 AtoFloat

Description Convert an ascii floating point numeric string into the RAX/EAX register

Arguments

- Address of the floating point numeric string

Return Value(s)

- RAX/EAX will contain the floating point value

Example

```
.data
    myString db ”-123.45”, 0h
.bss
.code
    push myString
    call AtoFloat
```

Usage Notes

- The ascii string must be zero/null terminated
- The carry flag will be set if an invalid ascii floating point string is provided

---

3.5.6 UInttoA

Description Convert an unsigned integer to a Decimal ascii string

Arguments

- The double-word/quad-word unsigned integer to convert
- The address of an array to contain the ascii string

Return Value(s)

- None

Example

```
.data
    theNumber dd 12345 ;64-bit: theNumber dq 12345
.bss
    asciiString resb 12
.code
    push DWORD [theNumber] ;64-bit: push QWORD [theNumber]
    push asciiString
    call UInttoA
```

Usage Notes

- The ascii string will end with a null character, so must be long enough to hold the entire string

---

3.5.7 SInttoA

Description Convert an signed integer to a Decimal ascii string

Arguments

- The double-word/quad-word signed integer to convert
- The address of an array to contain the ascii string

Return Value(s)

- None

Example

```
.data
    theNumber dd -12345 ;64-bit: theNumber dq -12345
.bss
    asciiString resb 12
.code
    push DWORD [theNumber] ;64-bit: push QWORD [theNumber]
    push asciiString
    call SInttoA
```

Usage Notes

- The ascii string will end with a null character, so must be long enough to hold the entire string

3.6 Array Manipulation Functions

3.6.1 selectUDSort/selectUQSort

Description Using select sort, sort an unsigned double/quad signed word array in ascending or descending order

Arguments

- Address of the double/Quad word array
- Number of items in the array
- 0h=ascending, 1h=descending

Return Value(s)

- RAX/EAX will contain the number of swaps done, or zero if the array is already in sorted order or the number of items sent to the function is 0 or less than 0

Example

```
.data
    theArray dq 10h, 100h, 5h, 22h
    .lengthof equ $-theArray
.bss
.code
    push theArray
    push theArray.lengthof
    push 0h
    call selectUQSort
```

Usage Notes

- None

3.6.2 selectSDSort/selectSQSort

Description Using select sort, sort a signed double/quad word array in ascending or descending order

Arguments

- Address of the double/Quad unsgined word array
- Number of items in the array
- 0h=ascending, 1h=descending

Return Value(s)

- RAX/EAX will contain the number of swaps done, or zero if the array is already in sorted order or the number of items sent to the function is 0 or less than 0

Example

```
.data
    theArray dq -10h, 100h, -5h, 22h
    .lengthof equ $-theArray
.bss
.code
    push theArray
    push theArray.lengthof
    push 0h
    call selectSQSort
```

Usage Notes

- None

3.7 Date and Time Functions

Note: These functions are not 100% accurate and still under development

3.7.1 GetCurrentHour

Description The GetCurrentHour function is used to obtain the current 24-hour formatted hour as obtained from the local computer's system clock

Arguments

- None

Return Value(s)

- RAX/EAX will contain the current hour

Example

```
    .data
    .bss
    .code
        call GetCurrentHour
```

Usage Notes

- None

3.7.2 GetCurrentMinute

Description The GetCurrentMinute function is used to obtain the current minute as obtained from the local computer's system clock

Arguments

- None

Return Value(s)

- RAX/EAX will contain the current minute

Example

```
    .data
    .bss
    .code
        call GetCurrentMinute
```

Usage Notes

- None

3.7.3 GetCurrentSecond

Description The GetCurrentSecond function is used to obtain the current second as obtained from the local computer's system clock

Arguments

- None

Return Value(s)

- RAX/EAX will contain the current second

Example

```
.data
.bss
.code
    call GetCurrentSecond
```

Usage Notes

- None

3.7.4 GetCurrentYear

Description The GetCurrentYear function is used to obtain the current year as obtained from the local computer's system clock

Arguments

- None

Return Value(s)

- RAX/EAX will contain the current year

Example

```
.data
.bss
.code
    call GetCurrentYear
```

Usage Notes

- None

3.7.5 GetCurrentMonth

Description The GetCurrentMonth function is used to obtain the current month as obtained from the local computer's system clock

Arguments

- None

Return Value(s)

- RAX/EAX will contain the current month

Example

```
.data
.bss
.code
    call GetCurrentMonth
```

Usage Notes

- None

3.7.6 GetCurrentDay

Description The GetCurrentDay function is used to obtain the current day as obtained from the local computer's system clock

Arguments

- None

Return Value(s)

- RAX/EAX will contain the current day

Example

```
.data
.bss
.code
    call GetCurrentDay
```

Usage Notes

- None

3.7.7 PrintSystemTime

Description The PrintSystemTime function is used to print the system time to StdOut

Arguments

- None

Return Value(s)

- None

Example

```
.data
.bss
.code
    call PrintSystemTime
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed

---

3.7.8 PrintSystemDateEng

Description The PrintSystemDateEng function is used to print the system date to StdOut in the format mm/dd/yyyy

Arguments

- None

Return Value(s)

- None

Example

```
.data
.bss
.code
    call PrintSystemDateEng
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed

---

3.7.9 PrintSystemDateEuro

Description The PrintSystemDateEuro function is used to print the system date to StdOut in the format yyyy/mm/dd

Arguments

- None

Return Value(s)

- None

Example

```
.data
.bss
.code
    call PrintSystemDateEuro
```

Usage Notes

- No carriage return or line feed will be printed after the string or sub-string is printed

---

3.7.10 GetEngDateString

Description The GetEngDateString function is used to get the system date in the format mm/dd/yyyy and put it into a string array

Arguments

- A Byte-buffer

Return Value(s)

- RAX/EAX will contain the number of characters returned

Example

```
.data
.bss
    datebuff resb 20
.code
    push datebuff
    call GetEngDateString
```

Usage Notes

- None

3.7.11 GetEuroDateString

Description The GetEuroDateString function is used to get the system date in the format yyyy/mm/dd and put it into a string array

Arguments

- A Byte-buffer

Return Value(s)

- RAX/EAX will contain the number of characters returned

Example

```
.data
.bss
    datebuff resb 20
.code
    push datebuff
    call GetEuroDateString
```

Usage Notes

- None

3.7.12 GetTimeString

Description The GetTimeString function is used to get the system time in the format hh:mm:ss and put it into a string array

Arguments

- A Byte-buffer

Return Value(s)

- RAX/EAX will contain the number of characters returned

Example

```
.data
.bss
    timebuff resb 20
.code
    push timebuff
    call GetTimeString
```

Usage Notes

- None