

[console, argument](#)

== [apollo_asm](#) ==

another example to get argument from cli at execution time, and then print on the console.

```

    include exec/types.i
    include exec/libraries.i
    include     exec/exec_lib.i
    include     dos/dos_lib.i
    include     dos/dos.i

    move.l     a0,string           ;this is not used in this prg... but I
leave that
    move.l     d0,size             ; this too

*----- opening DOS library -----
openlibs
    move.l     $4,_ExecBase
    lea       DosName,A1
    move.l     #0,d0
    CALLEXEC  OpenLibrary
    move.l     d0,_DOSBase
*----- getting stdinut and output -----
    CALLDOS   Output
    move.l     d0,_stdout
    CALLDOS   Input
    move.l     d0,_stdinut

*----- we will use three DOS lib functions -----
*----- ReadItem,StrToLong and VPrintf to read argument, convert to long
value
*----- and print to CLI. Pls looks at :
*_
http://amigadev.elowar.com/read/ADCD\_2.1/Includes\_and\_Autodocs\_3.\_guide/node01A2.html
*_
http://amigadev.elowar.com/read/ADCD\_2.1/Includes\_and\_Autodocs\_3.\_guide/node01C4.html
*_
http://amigadev.elowar.com/read/ADCD\_2.1/Includes\_and\_Autodocs\_3.\_guide/node01CE.html
*_ and
http://amigadev.elowar.com/read/ADCD\_2.1/Includes\_and\_Autodocs\_3.\_guide/node0198.html
*_
http://amigadev.elowar.com/read/ADCD\_2.1/Includes\_and\_Autodocs\_3.\_guide/node017F.html
*----- getting arguments -----
get_one_arg

```

```

    move.l    #arg_string,d1          ; move buffer ptr in d1
    move.l    #100,d2                ; move buffer size in d2
    clr.l     d3                     ; no input specified = read from
Input()
    CALLDOS ReadItem                 ; read one argument, now we have it in
our buffer
    jsr convertstr                   ; try convert arg to long value
get_next_arg
    move.l    #arg_string,d1          ; move buffer ptr in d1
    move.l    #100,d2                ; move buffer size in d2
    clr.l     d3                     ; no input specified = read from
Input()
    CALLDOS ReadItem                 ; read one argument, now we have it in
our buffer
    cmp.l     #ITEM_NOTHING,d0       ; check do user give 2nd argument
    beq      std_message             ; branch to standard text if he do
not do that
change_text
    move.l    #arg_string,d1          ; load pointer 2nd arg string into
d1
    move.l    #0,d2                  ; plain text without repeat number
print
    move.l    #1,d3                  ; loop counter
    jsr      repeat_text             ; print text
    bra      closelibs
std_message
    move.l    #text,d1
    move.l    #str_conv,d2
    move.l    #1,d3
    jsr      repeat_text

closelibs
    move.l    _DOSBase,A1
    CALLEXEC CloseLibrary
*----- end of prg -----
endprg  clr.l    d0
        RTS
*----- repeat text n times routine -----
*----- input values for VPrintf must be in d1 and d2 registers -----
repeat_text
    move.l    d1,d1copy
    move.l    d2,d2copy
.loop
    cmp.l     str_conv,d3
    bhi      closelibs
    move.l    d3,powt
    CALLDOS VPrintf
    add.l     #1,d3
    move.l    d1copy,d1

```

```

    move.l    d2copy,d2
    bra      .loop
    RTS

*----- convert string routine -----
*----- it checkout do given string can be converted to valid value
*----- and do that value is in valid range -----
convertstr
    move.l    #arg_string,d1
    move.l    #str_conv,d2
    CALLDOS  StrToLong
    cmp.l     #-1,d0
    beq       prt_error
    move.l    str_conv,d0
    cmp.l     #1,d0
    blo       prt_error
    cmp.l     #20,d0
    bhi       prt_error
    move.l    #1,d3
    RTS

*----- print errors routine -----
*----- add.l to stack are because that routine can be called
*----- from middle of another subroutine, you may comment it
*----- and watch what will happend -----
prt_error
    move.l    #mess_error,d1
    move.l    #0,d2
    CALLDOS  VPrintf
    add.l     #4,sp
    bra       closelibs
prt_error2
    move.l    #mess_error2,d1
    move.l    #0,d2
    CALLDOS  VPrintf
    add.l     #4,sp
    bra       closelibs

size          dc.l    0
string        dc.l    0
str_conv      dc.l    5432
powt          dc.l    1234
dlugosc       dc.l    0
d1copy        dc.l    0
d2copy        dc.l    0
text          dc.b    "You give as argument number %lu. Repeat no
%lu",10,0
mess_error     dc.b    "You do not give number or your number is not in
range 1-20 !",10,0
mess_error2    dc.b    " You do not specified any text to print
!",10,0
_ExecBase     dc.l    0

```

Last update:
2016/08/31 11:32 get_argument_then_print_on_console2 https://wiki.apollo-accelerators.com/doku.php/get_argument_then_print_on_console2

```
_DOSBase          dc.l      0
DosName dc.b      "dos.library",0
_stdout           dc.l      0
_stdinput         dc.l      0
arg_string        ds.b      100
```

From:
<https://wiki.apollo-accelerators.com/> - **Apollo Accelerators Public Wiki**

Permanent link:
https://wiki.apollo-accelerators.com/doku.php/get_argument_then_print_on_console2

Last update: **2016/08/31 11:32**

