math, console

== apollo_asm ==

this program will calculate prime numbers up to 200,
and will print results on console.

```
* calculate prime numbers up to 200, print results on console.
* credits pisklak for the code

        MACHINE MC68020
        include exec/types.i
        include exec/libraries.i
        include    exec/exec_lib.i
        include    dos/dos_lib.i
max_num EQU            200
LF      EQU            10                          ; Line feed
        lea            prime_array,a0              ; load prime_array  addr
indo a0
        move.l        #2,d1                       ; setup counter reg
fill_loop:
        move.l        d1,(a0)+                   ; write value to arry
        add.l        #1,d1                       ; increasing counter
        cmp.l        #max_num,d1                  ; checking do we are not at
the end of array
        bls            fill_loop                  ; if not then execute fill
loop again

        lea            prime_array,a0              ; reset array addr in a0
        move.l        #2,d2                       ; starting position in d2

main_loop:
        cmp.l        #max_num,d2                  ; check do we not at last
position
        bhi            openlibs
        move.l        -8(a0,d2.l*4),d0           ; we get our number from arry
        cmp.l        #$0,d0                       ; and we check against our
"throw" sign
        beq            check_next                  ; if it is throwed one then
move on to next
        move.l        d0,d1                       ; clone it in d1
throw_loop:
        add.l        d0,d1                       ; now we have in d1 next
multiply of our number
        cmp.l        #max_num,d1                  ; we check do we do not
exceed our range
        bhi            check_next                  ; if no then check next
number from array
        move.l        #0,-8(a0,d1.l*4)           ; mark 'throwed' sign
        bra            throw_loop                  ; throw next multiply
```

```
check_next:
        add.l        #1,d2                    ; move to next number
        bra            main_loop

*   ----------------------- write calculated values part ------------------
-------------
openlibs:
        lea         _DosName,a1                ; load lib mane to open
        move.l        #0,d0                    ; we open any version
        CALLEXEC    OpenLibrary                ; now we open DOS library
        move.l        d0,_DOSBase                ; and store its addres in
_DOSBase variable
                                              ; note that we do not check do
open libs succed
        CALLDOS        Output                    ; get output handler
        move.l        d0,_stdout                ; store it in variable
print_welcome:
        move.l        #w_msg,d1                ; we load pointer to our
welcome message
        move.l        #w_msg_arg,d2            ; pointer to arguments arry
        CALLDOS        VPrintf                    ; print our welcome
message
print_prime:
        lea            prime_array ,a2
        move.l        #2,d3                    ; our counter - this time we
do it slighty different
findloop:
        cmp.l        #0,(a2)                    ; compare our array value
with out 'throwed'ámark
        bne            print_value                ; in not throwed then
print that value
        add.l        #4,a2                    ; move to next array position
        add.l        #1,d3
        cmp.l        #max_num,d3                ; check do we not reach end
of range
        blo            findloop                ; if not then check next
value
        bra         printLF                    ; if yes then this is the end
of prg -> print LF and closelibs
print_value:
        move.l        #printvaluetext,d1        ; text to print
        move.l        a2,d2                    ; value to print
        CALLDOS        VPrintf                    ; print value
        add.l        #4,a2                    ; move to next position in
array
        add.l        #1,d3                    ; increase our counter
        cmp.l        #max_num,d3                ; are we in range ?
        blo            findloop                ; if yes go check next value
```

```
printLF:
        move.l        #prLFstr,d1               ; we print a LF at the end
        move.l        #0,d2                   ; no args for VPrintf
        CALLDOS        VPrintf                   ; print newline

closelibs
        move.l    _DOSBase,a1                   ; load lib base to close
        CALLEXEC CloseLibrary               ; close it

endprg      RTS

_ExecBase           dc.l        0
_DOSBase            dc.l        0
_DosName            dc.b        "dos.library",0
_stdout             dc.l        0

w_msg               dc.b        LF," -* This program calculates and prints
prime numbers with range %.ld with sieve method *-",LF,0
w_msg_arg           dc.l        max_num
printvaluetext      dc.b        "%lu ",0
prLFstr             dc.b        LF,0
prime_array         DS.l        max_num
```