Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки Кафедра технічної кібернетики

Звіти до комп'ютерних практикумів з модуля «Системне програмування»

Прийняв доцент кафедри ТК Лісовиченко О.І. "…"2018 р. Виконав Студент групи IT-61 Лесогорський К.С.

Комп'ютерний практикум №2

Тема: засоби обміну даними.

Завдання:

- 1. Скласти процедуру введення і перетворення цілого числа.
- 2. Скласти і реалізувати програму введення та виведення цілого числа зі знаком та виведення рядка символів.

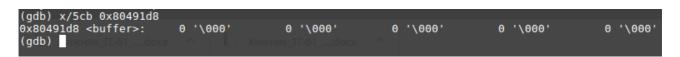
Текст програми:

```
.data
section
      sys_write equ 4h
      sys_read equ 3h
      std in equ 2h
      std_out equ 1h
      max chars equ 5h
      num dw -135
      msg db 'Not a valid character or too big/small number ', 0xa
      len equ $-msg
section
              .bss
      buffer resb 64
section
         .text
                                 ;must be declared for linker (ld)
global
         start
start:
      call read_num
      push ax
      call print_num
      sub esp,4 ;clear stack
      mov ax,1
      int 80h
print_num:;expects number on stack
      push ebp; save base pointer
      mov ebp,esp; point to start of stack
      res sign:
      mov ax, word [ebp+8]; ebp+ret address
      or ax,ax
      ins init ns
      mov ax, '-';print minus sign
      mov [buffer], ax
```

```
mov eax, sys_write
mov ebx, std out
mov ecx, buffer
mov edx, 1
int 80h
mov ax, [ebp+8];invert value
neg ax
mov [ebp+8], ax
jmp init_s
init_ns:
mov ax, [ebp+8]
init_s:
mov bx, 10
mov dx, 0
divide_loop:
add dx,2;increment counter by two, since two bytes will be reserved on stack
div bl
xor cx, cx
mov cl,ah
add cx, '0'
push cx
xor ah,ah
test ax,ax
jnz divide_loop
print_res:
mov ax, sys_write
mov bx, std_out
mov ecx, esp; counter is already in dx
int 80h
mov esp,ebp;restore stack and return
pop ebp
ret
read_num:
push ebp
mov ebp,esp
mov eax, sys_read
mov ebx, std in
mov ecx, buffer
```

```
mov edx, max_chars
int 80h
mov ecx, buffer; pointer
check_first_symbol:
mov al,[buffer]
cmp ax, '-'
je resolve_minus
mov ax,1
push ax
cmp ax, '+'
je resolve_plus
jmp preprocess_integer
resolve minus:
inc ecx
jmp preprocess_integer
resolve plus:
mov ax,0
push ax
inc ecx
jmp preprocess_integer
preprocess_integer:
mov bl,10; set first nul
push word 0; push empty value
process_integer:
mov al, [ecx]
sub ax, '0'
validate num:
is bad; if ax<0 - error
mov dx, 9
sub dx, ax
is bad
good:
mov dx,ax
pop ax
mul bl
jo bad
add ax,dx
jo bad; if overflow - it's time to leave
push ax
inc ecx
cmp [ecx],byte 0xa
```

```
je finish
      cmp [ecx], byte 0x0
      je finish
      jmp process_integer
      bad:
      mov ax,sys_write
      mov bx,std_out
      mov ecx, msg
      mov dx,len
      int 80h
      mov eax,1
      int 80h
      finish:
      pop ax
      pop bx
      cmp bx,1
      jne return
      neg ax
      return:
      mov esp, ebp
      pop ebp
      ret
Значення у секціях:
До введення числа:
```



Після введення числа -25:

(gdb) x/5bc 0x80491d8 0x8049<u>1</u>d8 <buffer>: 45 '-' 50 '2' 53 '5' 10 '\n' 0 '\000' K KOMAH (gdb)

Стан регістру ах після введення числа:

(gdb)	ira	ax							
ax (gdb)	nd	0xff	e7 -2	5					
(gub)	of 6		482 words, 3,031 c		aracters	Default Style			

Висновок:

- 1. В даному комп'ютерному практикумі було написано програму, яка включає процедуру для введення та перетворення з рядка в число цілого числа.
- 2. Також є процедура перевірки на неприпустимі символи.
- 3. І процедура на виведення цілого числа зі знаком.