Міністерство освіти і науки України Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Навчально-науковий Фізико-технічний інститут

Архітектура комп'ютерних систем Комп'ютерний практикум Робота №5

> Виконав: студент групи ФІ-12 Завалій Олександр Перевірив: Козленко О.В.

Робота №5.

Системні виклики в архітектурі х86 на асемблері для операційної системи Linux

Мета:

Ознайомитися з роботою системних викликів на мові асемблер в середовищі операційної системи Linux.

Варіант №4

Зміст індивідуального завдання:

Режимом доступу до файлу	Елементи	Операція	Системне
	масиву	з масивом	переривання
rw-r-rwx	8, 5, 3, 6, 7, 12	Ділення суми елементів на 2	Створити анонімний файл

Code

```
global _start
section .data
    array db 8,5,3,6,7,12
    arraylen equ 6
   msg db 'My array sum is: ', 0
    msglen equ $-msg
    filename db 'lab5_4', 0
    fract_part db '.5', 0
    fractlen equ $-fract_part
section .bss
    buffer resb 20
section .text
itoa:
   push ebp
    mov ebp, esp
    sub esp, 8
    mov eax, [ebp+8]
    lea edi, [buffer+20]
    mov ecx, 10
   mov dword [ebp-8], 0
.divloop:
   xor edx, edx
    idiv ecx
    add edx, '0'
    dec edi
    mov [edi], dl
    inc dword [ebp-8]
    cmp eax, 0
    jnz .divloop
    mov eax, edi
    mov ecx, [ebp-8]
    leave
    ret
_start:
   mov eax, arraylen
    xor ebx, ebx
    lea ecx, array
    xor edi, edi
startloop:
    cmp ebx, eax
    je endloop
    xor edx, edx
    mov dl, [ecx + ebx]
```

```
add edi, edx
    inc ebx
    jmp startloop
endloop:
   mov eax, edi
   mov edx, 0
    mov ecx, 2
    div ecx
   push eax
   call itoa
   add esp, 4
   mov eax, 4
   mov ebx, 1
   mov ecx, msg
    mov edx, msglen
    int 0x80
    mov eax, 4
    mov ebx, 1
    mov ecx, buffer
   mov edx, 20
    int 0x80
   cmp edx, 0
    je skip_fraction
    mov eax, 4
    mov ebx, 1
    mov ecx, fract_part
    mov edx, fractlen
    int 0x80
skip_fraction:
   mov eax, 8
   mov ebx, filename
   mov ecx, 00600
    int 0x80
   mov eax, 15
    mov ebx, filename
    mov ecx, 0o647
    int 0x80
   mov eax, 356
    mov ebx, filename
    mov ecx, 1
    int 0x80
```

```
mov eax, 1
xor ebx, ebx
int 0x80
```

Results

За допомогою утиліти strace можемо відслідкувати взаємодію нашої програми та ядра лінукс. В результаті бачимо рядок memfd_create("lab5_4 MFD_CLOEXEC) = 4. Розберемо параметри:

- memfd_create створює анонімний файл у пам'яті із назвою lab5_4 та прапором MFD CLOEXEC. Цей файл буде автоматично закрито при виконанні **execve**.
- $\bullet = 4 файловий дескриптор файлу.$

Анонімний файл, який ми створили за допомогою **memfd_create** не має імені у файловій системі. Проте має файловий дескриптор, який повертається системним викликом у регістрі **eax**. Цей дескриптор можемо використовувати для операцій з файлом.