Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №8 по курсу**

**«Операционные системы»**

Группа: М8О-203Б-23

Студент: Никитцев А.В.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 25.12.24

Москва, 2024

**Постановка задачи**

При выполнении лабораторных работ по курсу ОС необходимо продемонстрировать ключевые системные вызовы, которые в них используются и то, что их использование соответствует варианту ЛР

**Протокол выполнения**

**Running program:** ./lab\_4\_static

Enter command: 1

Enter n: 10

Result: 3.0677

Enter command: 2

Enter n: 10

Result: 3.04184

Enter command: 3

Enter size: 15

Enter values separated by space: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Sorting...

Result: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Enter command: 3

Enter size: 3

Enter values separated by space: 3 2 1

Sorting...

Result: 1 2 3

Enter command: ^C

**Strace Result:**

strace -p 189377

strace: Process 189377 attached

read(0, "1\n", 1024) = 2

write(1, "Enter n: ", 9) = 9

read(0, "10\n", 1024) = 3

write(1, "Result: 3.0677\n", 15) = 15

write(1, "Enter command: ", 15) = 15

read(0, "2\n", 1024) = 2

write(1, "Enter n: ", 9) = 9

read(0, "10\n", 1024) = 3

write(1, "Result: 3.04184\n", 16) = 16

write(1, "Enter command: ", 15) = 15

read(0, 0x55e5cea032c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, "3\n", 1024) = 2

write(1, "Enter size: ", 12) = 12

read(0, "15\n", 1024) = 3

write(1, "Enter values separated by space:"..., 33) = 33

read(0, "1 2 3 4 5 6 7 8 9 10 11 12 13 14"..., 1024) = 36

write(1, "Sorting...\n", 11) = 11

write(1, "Result: 1 2 3 4 5 6 7 8 9 10 11 "..., 45) = 45

write(1, "Enter command: ", 15) = 15

read(0, "3\n", 1024) = 2

write(1, "Enter size: ", 12) = 12

read(0, "3\n", 1024) = 2

write(1, "Enter values separated by space:"..., 33) = 33

read(0, "3 2 1\n", 1024) = 6

write(1, "Sorting...\n", 11) = 11

write(1, "Result: 1 2 3 \n", 15) = 15

write(1, "Enter command: ", 15) = 15

read(0, "", 1024) = 0

brk(0x55e5cea34000) = 0x55e5cea34000

mmap(NULL, 135168, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512d0f6000

brk(0x55e5cea25000) = 0x55e5cea25000

mmap(NULL, 266240, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512d0b5000

munmap(0x7f512d0f6000, 135168) = 0

mmap(NULL, 528384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512d034000

munmap(0x7f512d0b5000, 266240) = 0

mmap(NULL, 1052672, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512cf33000

munmap(0x7f512d034000, 528384) = 0

mmap(NULL, 2101248, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512cd32000

munmap(0x7f512cf33000, 1052672) = 0

mmap(NULL, 4198400, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512c931000

munmap(0x7f512cd32000, 2101248) = 0

mmap(NULL, 8392704, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512c130000

munmap(0x7f512c931000, 4198400) = 0

mmap(NULL, 16781312, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512b12f000

munmap(0x7f512c130000, 8392704) = 0

mmap(NULL, 33558528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512912e000

munmap(0x7f512b12f000, 16781312) = 0

mmap(NULL, 67112960, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f512512d000

munmap(0x7f512912e000, 33558528) = 0

mmap(NULL, 134221824, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f511d12c000

munmap(0x7f512512d000, 67112960) = 0

mmap(NULL, 268439552, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f510d12b000

munmap(0x7f511d12c000, 134221824) = 0

--- SIGINT {si\_signo=SIGINT, si\_code=SI\_KERNEL} ---

+++ killed by SIGINT +++

**Заключение**

Подводя итог, можно сказать, что у каждой стратегии есть свои плюсы и минусы. Например алгоритм свободных блоков прост в реализации относительно алгоритма двойников, однако по перформансу в некоторых метриках от обгоняет свободные блоки.