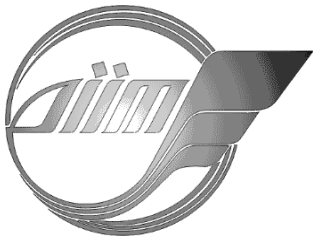
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



Дніпровський національний університет
залізничного транспорту імені академіка В. Лазаряна

Кафедра «Комп'ютерні інформаційні технології»

Лабораторна робота №3

з дисципліни «Операційні Системи»

на тему: «Купи. Робота з віртуальною пам'яттю»

Виконав:
студент гр.ПЗ1911
Сафонов Д.Є.
Прийняв:
Андрющенко В.О.

Дніпро, 2021

Тема. Купи. Робота з віртуальною пам'яттю.
Завдання. Завдання складається з трьох частин:

1) Скласти «карту пам’яті» - вивести інформацію про регіони адресного простору процесу (стан, доступ, розмір).
2) Перевірити дію атрибутів доступу до сторінок та гранулярність виділення пам’яті.
3) Створити додаткові купи процесу та використовувати їх, перевантаживши операції new та delete для динамічних структур даних.

Текст програми. github

Результати виконання програми.

```
[redacted] 1
sysinf =
        WORD      wProcessorArchitecture      = x64
        DWORD     dwPageSize                  = 0x1000
        LPVOID    lpMinimumApplicationAddress = 0x10000
        LPVOID    lpMaximumApplicationAddress = 0x7fffffeffff
        DWORD_PTR dwActiveProcessorMask       = 4095
        DWORD     dwNumberOfProcessors        = 12
        DWORD     dwAllocationGranularity     = 0x10000
        WORD      wProcessorLevel             = 23
        WORD      wProcessorRevision          = 28928
pmc =
        DWORD  PageFaultCount             = 0
        SIZE_T PeakWorkingSetSize         = 0xa93000
        SIZE_T WorkingSetSize             = 0xa93000
        SIZE_T QuotaPeakPagedPoolUsage    = 0
        SIZE_T QuotaPagedPoolUsage        = 0
        SIZE_T QuotaPeakNonPagedPoolUsage = 0
        SIZE_T QuotaNonPagedPoolUsage     = 0
        SIZE_T PagefileUsage              = 0x743000
        SIZE_T PeakPagefileUsage          = 0x743000
mbi0 =
        DWORD  AllocationProtect = no access
        SIZE_T RegionSize        = 0x10000
        DWORD  State             = MEM_FREE
        DWORD  Protect           = PAGE_NOACCESS
mbi1 =
        DWORD  AllocationProtect = PAGE_READONLY
        SIZE_T RegionSize        = 0xf000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READONLY
mbi2 =
        DWORD  AllocationProtect = no access
        SIZE_T RegionSize        = 0x1000
        DWORD  State             = MEM_FREE
        DWORD  Protect           = PAGE_NOACCESS
mbi3 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x1000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_NOACCESS
mbi4 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x1000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE & PAGE_GUARD
mbi5 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x1fe000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE
mbi6 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x20000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE
mbi7 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x10000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE
mbi8 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x100000
        DWORD  State             = MEM_RESERVE
        DWORD  Protect           = no access
mbi9 =
        DWORD  AllocationProtect = PAGE_READONLY
        SIZE_T RegionSize        = 0x334000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READONLY
testing page with protection: PAGE_READONLY
read successfull
[redacted] Unhandled page fault on write access to 0000000000010000 at address 0000000140003EB7 (thread 0104), starting debugger...
```

Рисунок 1: Перевірка сторінки із доступом для читання

```
[redacted] 2
sysinf =
        WORD      wProcessorArchitecture      = x64
        DWORD     dwPageSize                  = 0x1000
        LPVOID    lpMinimumApplicationAddress = 0x10000
        LPVOID    lpMaximumApplicationAddress = 0x7fffffeffff
        DWORD_PTR dwActiveProcessorMask       = 4095
        DWORD     dwNumberOfProcessors        = 12
        DWORD     dwAllocationGranularity     = 0x10000
        WORD      wProcessorLevel             = 23
        WORD      wProcessorRevision          = 28928
pmc =
        DWORD  PageFaultCount             = 0
        SIZE_T PeakWorkingSetSize         = 0xaad000
        SIZE_T WorkingSetSize             = 0xaad000
        SIZE_T QuotaPeakPagedPoolUsage    = 0
        SIZE_T QuotaPagedPoolUsage        = 0
        SIZE_T QuotaPeakNonPagedPoolUsage = 0
        SIZE_T QuotaNonPagedPoolUsage     = 0
        SIZE_T PagefileUsage              = 0x743000
        SIZE_T PeakPagefileUsage          = 0x743000
mbi0 =
        DWORD  AllocationProtect = no access
        SIZE_T RegionSize        = 0x10000
        DWORD  State             = MEM_FREE
        DWORD  Protect           = PAGE_NOACCESS
mbi1 =
        DWORD  AllocationProtect = PAGE_READONLY
        SIZE_T RegionSize        = 0xf000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READONLY
mbi2 =
        DWORD  AllocationProtect = no access
        SIZE_T RegionSize        = 0x1000
        DWORD  State             = MEM_FREE
        DWORD  Protect           = PAGE_NOACCESS
mbi3 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x1000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_NOACCESS
mbi4 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x1000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE & PAGE_GUARD
mbi5 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x1fe000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE
mbi6 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x20000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE
mbi7 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x10000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE
mbi8 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x100000
        DWORD  State             = MEM_RESERVE
        DWORD  Protect           = no access
mbi9 =
        DWORD  AllocationProtect = PAGE_READONLY
        SIZE_T RegionSize        = 0x334000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READONLY
testing page with protection: PAGE_READWRITE
read successfull
write successfull
Tried to allocate 0 bytes, got 0x10000 bytes
```

Рисунок 2: Перевірка сторінки із доступом для читання та запису

```
[redacted] 3
sysinf =
        WORD      wProcessorArchitecture      = x64
        DWORD     dwPageSize                  = 0x1000
        LPVOID    lpMinimumApplicationAddress = 0x10000
        LPVOID    lpMaximumApplicationAddress = 0x7fffffeffff
        DWORD_PTR dwActiveProcessorMask       = 4095
        DWORD     dwNumberOfProcessors        = 12
        DWORD     dwAllocationGranularity     = 0x10000
        WORD      wProcessorLevel             = 23
        WORD      wProcessorRevision          = 28928
pmc =
        DWORD  PageFaultCount             = 0
        SIZE_T PeakWorkingSetSize         = 0xa91000
        SIZE_T WorkingSetSize             = 0xa91000
        SIZE_T QuotaPeakPagedPoolUsage    = 0
        SIZE_T QuotaPagedPoolUsage        = 0
        SIZE_T QuotaPeakNonPagedPoolUsage = 0
        SIZE_T QuotaNonPagedPoolUsage     = 0
        SIZE_T PagefileUsage              = 0x744000
        SIZE_T PeakPagefileUsage          = 0x744000
mbi0 =
        DWORD  AllocationProtect = no access
        SIZE_T RegionSize        = 0x10000
        DWORD  State             = MEM_FREE
        DWORD  Protect           = PAGE_NOACCESS
mbi1 =
        DWORD  AllocationProtect = PAGE_READONLY
        SIZE_T RegionSize        = 0xf000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READONLY
mbi2 =
        DWORD  AllocationProtect = no access
        SIZE_T RegionSize        = 0x1000
        DWORD  State             = MEM_FREE
        DWORD  Protect           = PAGE_NOACCESS
mbi3 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x1000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_NOACCESS
mbi4 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x1000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE & PAGE_GUARD
mbi5 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x1fe000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE
mbi6 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x20000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE
mbi7 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x10000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READWRITE
mbi8 =
        DWORD  AllocationProtect = PAGE_READWRITE
        SIZE_T RegionSize        = 0x100000
        DWORD  State             = MEM_RESERVE
        DWORD  Protect           = no access
mbi9 =
        DWORD  AllocationProtect = PAGE_READONLY
        SIZE_T RegionSize        = 0x334000
        DWORD  State             = MEM_COMMIT
        DWORD  Protect           = PAGE_READONLY
testing page with protection: PAGE_NOACCESS
[redacted]  Unhandled page fault on read access to 0000000000000000 at address 0000000140003E83 (thread 0108), starting debugger...
```

Рисунок 3: Перевірка сторінки без доступу

Аналіз результатів та висновки.

Розроблена програма перевантажує оператори виділення та звільнення пам'яті, виводить інформацію про перші 10 сторінок пам’яті процесу та перевіряє атрибути доступу до сторінок і гранулярність виділення пам'яті. Як можна побачити з прикладів за спроби виконати недозволений доступ до пам’яті виникає виключення, а розмір виділеної пам'яті насправді округляється до гранулярності.