

Лабораторна робота №2 (семестр 2)

ТЕМА: Механізм керування пам'яттю

МЕТА: Отримати практичні навички використання функцій Win API.

ЗНАТИ: Основи програмування в ОС Windows.

ВМІТИ: Застосовувати API-функції.

ТЕОРЕТИЧНІ ВІДОМОСТІ.

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

ЗАВДАННЯ

Використовуючи наявну електронну документацію повністю повторити функціонал програм наведених в прикладах на мові C++ чи C#.

Тобто провести перетворення програм (трансляцію), представлених на мові програмування Object Pascal на мову C++ чи C#.

ПРИКЛАДИ

“1. VirtualMemState” – дослідження структури адресного простору процесу.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Для чого використовується диспетчер віртуальної пам'яті?
2. Який обсяг віртуального адресного простору на 32-бітних системах?
3. Опишіть структуру віртуального адресного простору процесу.
4. Який обсяг віртуальної пам'яті доступний користувачькому коду на 32-бітних системах?
5. Яке призначення розділу виявлення нульових вказівників?
6. Як прикладній програмі можна забезпечити максимальний обсяг доступної віртуальної пам'яті?
7. Опишіть схему використання віртуальної пам'яті у прикладній програмі.
8. Для чого використовується функція *VirtualAlloc*?
9. У чому полягає гранулярність виділення пам'яті? Які причини гранулярності?
10. Як можна виділити певний обсяг пам'яті за конкретно заданою адресою?
11. Як зарезервувати регіон адресного простору?
12. Яким чином можна отримати розмір сторінки пам'яті?
13. Як здійснюють проєкцію фізичної пам'яті на віртуальну?
14. Як у програмі виділити 500 МБ пам'яті та заповнити їх даними?
15. Як вивільняють раніше виділену пам'ять?
16. Для чого використовують сторінковий файл?
17. Чи може система працювати без сторінкового файлу?
18. Яка мінімальна порція даних має індивідуальні атрибути захисту?
19. Які атрибути захисту пам'яті ви знаєте?
20. Для чого використовують функцію *VirtualFree*?
21. Як змінюють атрибути захисту регіону сторінок віртуальної пам'яті?
22. Як задати атрибути захисту певного байту віртуальної пам'яті?
23. Для чого призначена функція *VirtualQuery*?
24. Чи можна отримати інформацію про зайнятість віртуальної пам'яті стороннього процесу?
25. У чому полягають недоліки безпосереднього використання віртуальної пам'яті для невеликих масивів даних?
26. Яке призначення механізму куп?
27. Що таке купа?
28. Яким чином система забезпечує потокобезпечність куп?
29. Яка особливість середовища Delphi при роботі з купами?
30. Чи може Delphi-програма використовувати стандартний механізм куп?