

Лабораторная работа по курсу «Системное программирование»

ТЕМА: УПРАВЛЕНИЕ ПРОЦЕССАМИ В WIN32 API

Цель работы:

Изучение вопросов управления процессами в WIN API. Получение практических навыков разработки приложений, выполняющих запуск дочерних процессов, управление процессами, получение информации о процессах. Получение навыков организации взаимодействия процессов и создания многопроцессных комплексов.

Задание:

Изучить управление процессами в Win32 API.

Приобрести практические навыки применения функций Win32 API для управления процессами в Windows-приложении и обеспечения взаимодействия процессов.

В качестве практической части работы необходимо создать приложение, которое будет обеспечивать создание дочерних процессов с запуском приложений Notepad.exe, Notepad.exe с открытием указанного файла, Calc.exe, собственного приложения. Для дочерних процессов необходимо предусмотреть возможность получения и отображения различной информации состояния.

Теоретические сведения:

Необходимая информация по теме лабораторной работы представлена:

- 1) в материалах лекции [1];
- 2) в главе 4 книги Джеффри Рихтера и Кристофа Назара "Windows via C/C++ " [2];
- 3) в главе 4 книги Рихтера "Windows для профессионалов. Создание эффективных WIN32-приложений с учетом специфики 64-разрядной версии Windows" (4-е издание) [3];
- 3) в разделе Platform SDK: DLLs, Processes, and Threads MSDN [3].

Дополнительные сведения представлены в главе 1 (обработка ошибок), главе 2 (работа с Unicode) и главе 3 (объекты ядра) книги [2] и [3].

Выполнение:

1. В своей рабочей папке создайте проект типа Win32 Application:

Создайте типовой проект с главным окном или воспользуйтесь проектом предыдущей работы по диалоговым окнам.

Назовите проект **SP_LB2-1. Строка заголовка главного окна должна отображать название работы, фамилию и инициалы слушателя.**
Например, "SP-LB2-1 Сидоров Д.В. Гр. 00321-2".

Выполните компиляцию, компоновку и запустите программу на выполнение. Убедитесь в работоспособности программы.

2. Добавьте в приложение код и ресурсы для создания дочерних процессов:

2.1 Изменения в главном меню

В главное меню добавьте:

а) меню *“Процессы”* с командами *“Блокнот”*, *“Блокнот с текстом”*, *“Калькулятор”*, *“Закрыть калькулятор”*;

б) меню *“Информация о процессах”* с командами *“Текущий процесс(0)”*, *“Блокнот (1)”*, *“Блокнот с текстом (2)”*, *“Калькулятор (3)”*.

2.2 Организация данных

Создайте глобальные массивы:

HANDLE ProcHandle[4] для дескрипторов процессов;

DWORD ProcId[4] для идентификаторов процессов;

HANDLE ThreadHandle[4] для дескрипторов потоков;

DWORD ThreadId[4] для идентификаторов потоков;

LPTSTR ProcImage[4] для указателей строк, идентифицирующих файлы запускаемых программ;

TCHAR CmdParam[4][260] для строк с параметрами запускаемых программ.

Предусмотрите код, обеспечивающий заполнение массивов ProcImage и CmdParam необходимой информацией. Индекс 0 предназначен для текущего процесса, индекс 1 – для процесса «Блокнот» и т.д.

2.3 Изменения в коде программы

Добавьте в приложение код обработки новых команд меню с выполнением следующих действий:

а) Запуск приложения «Блокнот».

По команде *“Процессы - Блокнот”* создается новый процесс с запуском приложения *Notepad.exe*. Имя образа процесса и параметры

командной строки процесса необходимо брать из элементов массивов ProcImage и (или) CmdParam с индексом 1.

Полученные при успешном запуске процесса дескрипторы и идентификаторы сохраните в массивах ProcHandle, ThreadHandle, ProcId и ThreadId под индексом 1.

Закройте дескрипторы созданных процесса и потока, чтобы “отвязать” дочерний процесс от родительского приложения.

Выполните построение приложения и проверьте работу.

б) Запуск приложения «Блокнот» с загрузкой текстового документа.

По команде **“Процессы – Блокнот с текстом”** создается новый процесс с запуском приложения *Notepad.exe* и загрузкой в него текстового документа. В качестве документа возьмите файл исходного текста приложения. Имя образа процесса и параметры командной строки процесса необходимо брать из элементов с индексом 2 в массивах ProcImage и (или) CmdParam.

Полученные при успешном запуске процесса дескрипторы и идентификаторы сохраните в массивах ProcHandle, ThreadHandle, ThreadHandle, ThreadId под индексом 2.

Дескрипторы созданного процесса **не закрывать**.

Выполните построение приложения и проверьте работу.

в) Запуск программы Calc.exe.

По команде **“Калькулятор”** создать новый процесс и запустить в нем приложение *Calc.exe*.

Имя образа процесса и параметры командной строки процесса необходимо брать из элементов массивов ProcImage и (или) CmdParam с индексом 3.

Полученные при успешном запуске процесса дескрипторы и идентификаторы сохраните в массивах ProcHandle, ThreadHandle, ThreadHandle, ThreadId под индексом 3.

Не закрывать дескрипторы созданного процесса.

Выполните построение приложения и проверьте работу.

г) Завершение работы дочернего процесса.

По команде **“Заккрыть калькулятор”** обеспечьте прекращение работы программы калькулятора. Код завершения должен быть численно равен порядковому номеру в списке группы.

Выполните построение приложения и проверьте работу.

3. Добавьте в приложение ресурсы и код для получения информации состояния дочерних процессов

По команде **“Информация о процессах”** → **“XXXX”** в диалоговом окне выводить информацию о запущенном процессе XXXX:

- имя файла программы (модуля приложения) и параметры командной строки;
 - дескриптор и идентификатор процесса;
 - дескриптор и идентификатор первичного потока;
 - код завершения процесса (запросить вызовом `GetExitCodeProcess`);
 - код завершения потока (запросить вызовом `GetExitCodeThread`);
- Если соответствующий процесс или поток не завершились, выводить вместо значения кода завершения строку **“Состояние - Активен”**;
- идентификатор класса приоритета процесса (запросить вызовом `GetPriorityClass`);
 - временные характеристики:
 - а) время жизни процесса;
 - б) время выполнения в режиме пользователя;
 - в) время выполнения в режиме ядра;
 - г) время простоя.
- (использовать системные вызовы `GetTickCount` и `GetProcessTimes`);

Примечания: 1. Для вывода используйте модальное диалоговое окно на базе шаблона с элементами класса `Static` для подписей и элементами класса `Edit` для отображения параметров процессов;

2. Чтобы использовать одного диалогового окна для вывода информации о разных процессах создавайте его с помощью вызовов **`DialogBoxParam`** и передавайте номер соответствующего процесса в качестве последнего параметра вызова. Получение и вывод информации о процессе выполняйте в обработчике сообщения **`WM_INITDIALOG`** оконной процедуры диалогового окна.

Используйте параметр **IParam** в качестве индекса для обращения к массивам ProcHand, ThreadHand, ProcId, ThreadId, ProcImage и CmdParam с целью получения постоянной информации о процессе. Для получения динамической информации состояния используйте соответствующие системные вызовы и не забывайте контролировать корректность их выполнения.

Процесс номер 0 - это текущий процесс.

Выполните построение приложения и проверьте работу.

4. Выполните эксперименты с полученной программой:

4.1 Создать процесс 1 и посмотреть информацию о нем.

Закрыть запущенный Блокнот и посмотреть информацию о процессе 1;

4.2 Создать процесс 2 и посмотреть информацию о нем.

4.3 Закрыть окно Блокнота с текстом файла Test.txt.

Посмотреть информацию о процессе 2.

4.4 Создать процесс 3 и посмотреть информацию о нем.

4.5 Выполнить команду “Закрыть калькулятор” и посмотреть информацию о процессе 3.

Контролировать получаемую информацию о процессах с помощью программы Process Viewer из пакета инструментальных средств Microsoft Visual Studio и программы «Process Explorer».

Дать пояснения к полученной информации.

5. Задание по самостоятельной работе:

5.1 Модифицируйте подпункт б) пункта 2.3 таким образом, чтобы можно было выбирать любой текстовый файл для загрузки в блокнот;

5.2 Создайте свое приложение с именем TestProc. Добавьте в приложение **SP_LB2-1** команду меню **TestProc** и обеспечьте запуск приложения **TestProc** с возможностью управления размерами, положением и режимом отображения его главного окна (используйте структуру STARTUPINFO в родительском процессе);

5.3 Создайте приложение запускающее блокнот с открытием текстового файла, ожиданием пока пользователь закроет блокнот и последующим выводом содержимого текстового файла.

6. Оформите отчет о работе.

Отчет о самостоятельной работе оформить отдельно

ЛИТЕРАТУРА

Основная литература:

- 1 Материалы к лекции в папке курса и лабораторной работы
- 2 Джеффри Рихтер, Кристоф Назар. Windows via C/C++ / Пер. с англ.- М.: Издательство “Русская Редакция”; СПб. : Питер; 2008, ISBN: 978-5-7502-0367-3. Глава 4.
- 3 Джеффри Рихтер. Windows для профессионалов: создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows/ Пер. с англ.- 4-е изд. -СПб:Питер; М.: Издательско-торговый дом “Русская Редакция”, 2004, ISBN: 5-272-00384-5. Глава 4.
- 4 MSDN_2001, раздел Platform SDK: DLLs, Processes, and Threads
Электронный ресурс