МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Костромской государственный университет»

(КГУ)

ИАСТ

(наименование института)

Кафедра автоматизированных систем и технологий

(наименование кафедры)

09.03.02

Направление подготовки/Специальность Информационные системы и технологии

(наименование направления подготовки/специальности)

Дисциплина Архитектура ЭВМ

(наименование дисциплины)

Лабораторная работа №9.

Службы Windows.

Выполнили студенты Копосов Лев Владимирович

Копосов Владимир Владимирович

(фамилия, имя, отчество)

Группа 22-ИСбо-1б

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_

Кострома

Вопросы:

1. Для каких целей обычно используются службы в Windows?

Службы в Windows обычно используются для выполнения различных задач в фоновом режиме, таких как управление сетью, обновление программного обеспечения, обеспечение безопасности и т.д. Они могут быть запущены автоматически при запуске операционной системы или по запросу пользователя. Службы обеспечивают непрерывную работу операционной системы и ее компонентов, не привлекая внимания пользователя, но предоставляя свои возможности для других программ. В основном это драйверы и другие приложения. Службы подходят для работы с серверным окружением.

1. Какие сведения хранит менеджер служб для каждой службы?

Менеджер служб хранит следующие сведения для каждой службы:

1. Type: выполняется ли служба в отдельном процессе, или же совмещает процесс с другими.
2. Start: режим запуска: при загрузке, по требованию, отключено.
3. ErrorControl: насколько критичным является отказ этой службы, и что должен сделать диспетчер в случае отказа.
4. ImagePath: полный путь к исполняемому файлу службы, а также параметры командной строки для её запуска.
5. DependOnService: список имён служб, которые требуются для работы данной службы.
6. DisplayName: человеко-читаемое имя.
7. Description: описание службы, либо ссылка на ресурс, содержащий данную строку.
8. ObjectName: имя учетной записи, от имени которой данная служба должна выполняться.
9. Group: имя группы, к которой принадлежит данная служба.
10. В чём разница между обычным драйвером и kernel драйвером?

Разница между обычным драйвером и kernel драйвером заключается в следующем:

Обычный драйвер работает как приложение в кольце 3-его уровня от любого пользователя, то есть он запускается и работает в контексте пользовательского процесса. Он имеет ограниченный доступ к аппаратному обеспечению. Драйвер обычно используется для управления устройствами, которые не требуют прямого доступа к ядру операционной системы.

Драйвер ядра (kernel) работает в кольце 0-ого уровня в режиме ядра, что дает ему больше возможностей. У него есть прямой доступ к аппаратуре и полные привилегии при работе с ресурсами операционной системы. Драйвер обычно используется для задач, которые связаны с непосредственной работой операционной системы.

Вывод в консоль.

Задание 1.

Напишите программу, выводящую список текущих установленных служб и драйверов.

Алгоритм работы программы:

Откройте дескриптор менеджера служб вызвовом функции OpenSCManagerA().

Опишите буфер для получения сведений как vector<byte> buffer. Циклически вызывайте функцию EnumServicesStatusExA() и корректируйте размер буфера, пока её вызов не увенчается успехом, или пока GetLastError() не вернёт ошибку, отличную от ERROR\_MORE\_DATA.

Приведите адрес буфера (buffer.data()) к типу ENUM\_SERVICE\_STATUS\_PROCESSA\*. Используя выходной параметр lpServicesReturned, перечислите элементы полученного массива, и для каждого вызовите подпрограмму вывода сведений.

Закройте дескриптор менеджера служб вызовом CloseServiceHandle().

Алгоритм работы подпрограммы вывода сведений:

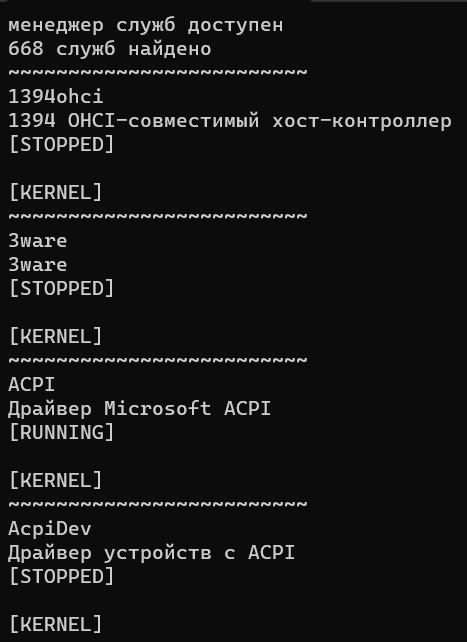
Подпрограмма должна принимать один параметр info - структуру типа ENUM\_SERVICE\_STATUS\_PROCESSA.

Выведите текущее состояние службы (info.ServiceStatusProcess.dwCurrentState). Используйте оператор switch-case.

Выведите тип службы (info.ServiceStatusProcess.dwServiceType). Так как тип является набором битовых флагов, нужно использовать побитовые операторы для сравнения значения с константами типов.

Выведите идентификатор службы (info.lpServiceName).

Выведите человеко-читаемое имя службы (info.lpDisplayName).



Задание 2.

Модифицируйте программу из задания 1 так, чтобы она выводила сведения о типе запуска службы, чувствительности к ошибкам, а также команду запуска и имя учётной записи (только для служб).

Измените тело программы следующим образом. Для каждого элемента массива служб она должна:

Открыть дескриптор службы вызовом OpenServiceA().

В случае успеха выделите буфер размером 8 КБ, и вызовите QueryServiceConfigA().

Интерпретируйте адрес буфера как адрес структуры вида QUERY\_SERVICE\_CONFIGA, и передайте его в подпрограмму вывода дополнительных сведений.

Закройте дескриптор службы, если он был успешно открыт.

Алгоритм работы подпрограммы вывода дополнительных сведений:

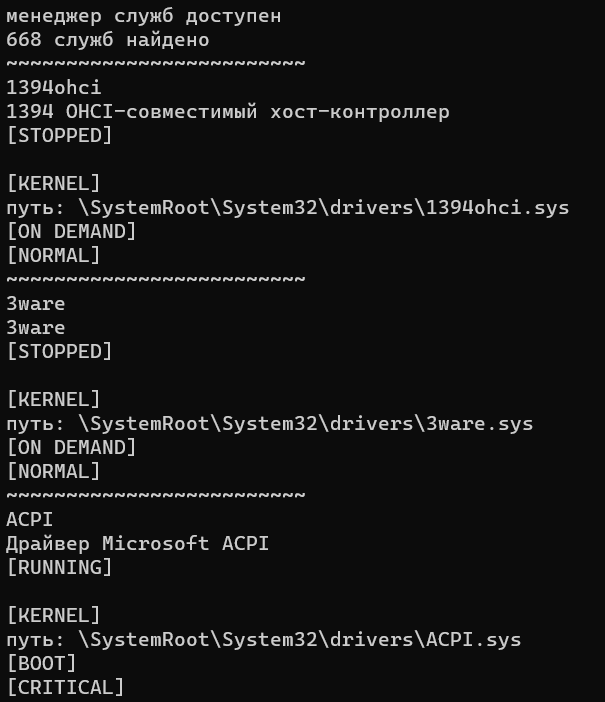
Подпрограмма должна принимать один параметр config типа QLPQUERY\_SERVICE\_CONFIGA.

Выведите тип запуска службы (config->dwStartType). Используйте оператор switch-case.

Выведите уровень чувствительности к ошибкам (config->dwErrorControl). Используйте оператор switch-case.

Выведите имя учётной записи пользователя (config->lpServiceStartName). Оно имеет смысл только если рассматриваемая служба не является драйвером. Проверьте поле config->dwServiceType.

Выведите команду запуска службы (config->lpBinaryPathName).



Дополнительное задание.

Модифицируйте программу из задания 2 следующим образом. Используя функцию QueryServiceConfig2A(), запросите человекочитаемое описание каждой службы(SERVICE\_CONFIG\_DESCRIPTION), и выведите его вместе с остальными данными. См. структуру SERVICE\_DESCRIPTIONA.

