**ЛЕКЦИЯ. Создание учетной записи пользователя**

Учетная запись создается для каждого пользователя компьютера. Она содержит логин, пароль, группу, роль и другую информацию о пользователе. Здесь мы рассмотрим, как ее создать программно. Для создания учетной записи пользователя используется функция

**NetuserAdd** с таким форматом

NET\_API\_STATUS NetUserAdd (

LPCWSTR servername, // имя сервера

DWORD level, // уровень информации

LPBYTE buf, // указатель на информацию об учетной записи

LPDWORD parm\_err // индексация структуры с информацией

);

Данную функцию могут выполнить только те пользователи, которые являются администраторами или операторами учетных записей.

В случае успешного завершения функция NetuserAdd возвращает

**NERR\_success**,

а в случае неудачи возможны следующие коды завершения:

□ error\_access\_denied — пользователю отказано в доступе;

□ NERR\_invaiidcomputer — неправильное имя компьютера;

□ NERR\_NotPrimary — операция может выполняться только на первичном

контроллере домена;

□ NERR\_GroupExists — группа уже существует;

□ NERR\_userExists — учетная запись уже существует;

□ NERR\_PasswordTooShort — пароль короче, чем требуется.

Параметр servername должен указывать на строку с именем сервера. Имя серевера должно иметь кодировку Unicode и начинаться с символов \\.

Параметр level указывает тип структуры с информацией о пользователе. Этот параметр принимать одно из следующих значений:

□ 1 — используется структура типа user\_info\_i;

□ 2 — используется структура типа user\_info\_2;

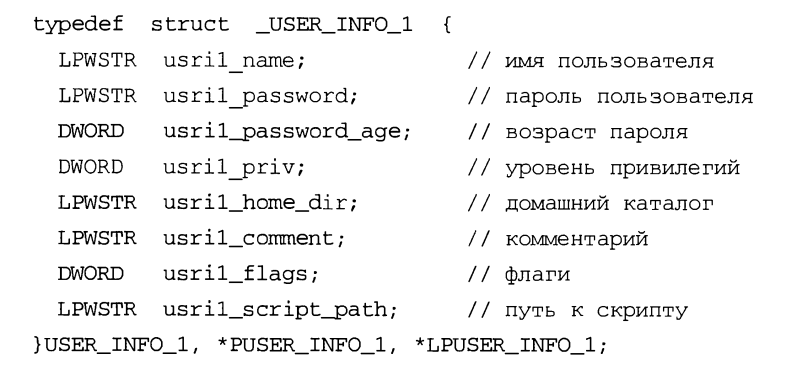
□ 3 — используется структура типа user\_info\_3;

□ 4 — используется структура типа user\_info\_4.

Параметр buf указывает на буфер с информацией о пользователе. Информация должна храниться в структуре, тип которой задан параметром level.

В параметр parm\_err функция NetuserAdd помещает индекс первого поля в структуре с информацией о пользователе, которое содержит неправильные данные и, как следствие, вызвало завершение функции NetuserAdd с ошибкой. В этом параметре может быть установлено значение null. В этом случае индекс поля с неправильными данными не возвращается.

Приведем структуру user\_info\_i, которая используется в примере. Эта структура имеет следующий тип:



В поле usrii\_name хранится указатель на строку с именем пользователя. Длина строки не должна превышать unlen байтов, а сама строка должна иметь кодировку Unicode.

В поле usrii\_password содержится указатель на строку на пароль. Длина строки не должна превышать pwlen байтов, а сама строка должна иметь кодировку Unicode.

Поле usrii\_password\_age задает количество секунд, которые истекли с момента последнего изменения пароля.

Поле usrii\_priv предназначено для хранения уровня привилегий, которые получает пользователь. Содержит одно из следующих значений:

□ USER\_PRIV\_GUEST — ГОСТЬ;

□ user\_priv\_user — пользователь;

□ user\_priv\_admin — администратор.

При вызове функции NetuserAdd это поле может содержать только значение

USER\_PRIV\_USER.

В поле usrii\_home\_dir хранится указатель на строку с именем домашнего каталога пользователя.

В поле usrii\_coiranent хранится указатель на строку с комментариями о пользователе. Это поле может содержать значение null.

Поле usrii\_fiags предназначено для хранения флагов, определяющих тип учетной записи и ее свойства. Используются такие флаги:

□ uf\_normal\_account — учетная запись обычного пользователя;

□ uf\_temp\_duplicate\_account — временная учетная запись, действительная

только в домене;

□ uf\_workstation\_trust\_account — учетная запись компьютера в домене

как рабочей станции, так и сервера;

□ uf\_server\_trust\_account — учетная запись вторичного контроллера

домена;

□ uf\_interdomain\_trust\_account — учетная запись в доверительном домене.

Кроме того, в этом поле может быть установлена любая комбинация следующих флагов:

□ uf\_script — исполнять скрипт (сценарий) при входе в систему;

□ uf\_accountdisable — учетная запись недействительна;

□ uf\_homedir\_required — требуется домашний каталог;

□ uf\_passwd\_notreqd — пароль не требуется;

□ uf\_passwd\_cant\_change — пароль нельзя изменять;

□ uf\_lockout — учетная запись заблокирована;

□ uf\_dont\_expire\_passwd — пароль не нужно изменять;

□ uf\_not\_delegated — учетная запись не может делегироваться другим

пользователям;

□ uf\_smartcard\_required — для входа в систему требуется смарт-карта;

□ uf\_use\_des\_key\_only — использовать только DES-стандарт шифрования;

□ uf\_dont\_require\_preauth — не требуется Kerberos-аутентификация при

входе в систему;

□ uf\_trusted\_for\_delegation — учетная запись может делегироваться;

□ uf\_password\_expired — время действия пароля истекло.

В поле usrii\_script\_path хранится указатель на строку с именем файла, содержащего скрипт, который выполняется перед входом пользователя в систему. Строка должна иметь кодировку Unicode. Это поле может содержать значение null.

Программа для создания учетной запись пользователя использует функциюNetUserAdd.

#include <stdio.h>

#include <windows.h>

#include <lm.h>

#pragma comment ( lib, "netapi32 .lib" ) // подключаем сетевую библиотеку

int main ()

{

wchar\_t s erver\_name[256] = L" \ \ \ \";

wchar\_t user\_name[UNLEN] ;

wchar\_t user\_j?assword[PWLEN] ;

wchar\_t user\_comment [256] ;

USER\_INFO\_l ui;

NET\_API\_STATUS ret\_s tatus;

printf (" Введи имя сервера: ") ;

// формируем имя сервера

wscanf (L"%s", server\_name + wcslen(server\_name) ) ;

// имя сервера

// имя пользователя

// пароль пользователя

// комментарии о пользователе

// информация о пользователе

// код возврата из функции

printf("Введи имя пользователя: "); // читаем имя пользователя

wscanf(L"%s", user\_name); // читаем имя пользователя

ui.usril\_name = user\_name; // устанавливаем имя пользователя

printf("Введи пароль: ") ;

wscanf(L"%s", user\_password); // читаем пароль пользователя

ui.usril\_password = user\_password; // устанавливаем пароль пользователя

ui.usril\_priv = USER\_PRIV\_USER; // обычный пользователь

ui.usril\_home\_dir = NULL; // домашнего каталога нет

printf("Input user comment: ") ;

getwchar(); // очищаем поток

\_getws (user\_comment) ; // читаем комментарии о пользователе

ui.usril\_comment = user\_comment; // устанавливаем комментарии

ui.usril\_flags = UF\_SCRIPT; // исполнять скрипт при входе

// пользователя в систему

ui.usril\_script\_path = NULL; // пока файл со скриптом не определяем

// добавляем пользователя

ret\_status = NetUserAdd(

server\_name, // имя сервера

1, // уровень информации 1

(LPBYTE)&ui, // адрес информации о пользователе

NULL); // индексирования в структуре данных нет

// проверяем на успешное завершение

if (ret\_status != NERR\_Success)

{

printf("Добавление уч. Записи пользователя завершилось неудачей.\n");

printf("Error code: %u\n", ret\_status);

return ret\_status;

}

printf("Уч. Запись пользователя добавлена\n");

return 0;