Topic 39: Setting File Times

UNIX Touch command changes file access and changes the time to current system time.

For changing file time the GetSystemTimeAsFileTime is more convenient than calling Get SystemTime followed by SystemTimeToFileTime.

These two functions are provided by sysinfoapi.h api used to retrieve system time.
GetSystemTimeAsFileTime() take file pointer as argument and retrieves the current system date and time in Coordinated Universal Time (UTC) format.

GetSystemTime() A pointer to a SYSTEMTIME structure to receive the current system date and time. Retrieves the current system date and time in Coordinated Universal Time (UTC) format.

SetFileTime() Sets the date and time that the specified file or directory was created, last accessed, or last modified.

Topic 40: File Processing Strategies

C Library for file processing have following features:

- Code is portable to Operating system running other than windows.
- Convenient line and character oriented functions are simply string processing.
- The functions are higher level and easy to use than windows functions.
- Line and stream character-oriented functions can be migrated to generic calls.

C library also have some limitations but some performance advantages are also present when compared to windows functions.

Topic 41: File Locking

Windows OS can lock files completely or some part of a file.There are two functions avalible LockfileEx and LockFile.

The syntax and parameters list of LockfileEx is given below:

```
BOOL LockFileEx(
        HANDLE hFile,
        DWORD dwFlags,
        DWORD dwReserved,
        DWORD nNumberOfBytesToLockLow,
        DWORD nNumberOfBytesToLockHigh,
        LPPVERLAPPED lpOverlapped
)
```

To Unlock a file UnlockFileEx() function is used.Parameters list is highly similar to LockFileEx.

Topic 42: Releasing file locks

If a program does not release a lock or holds the lock longer, other programs will not be able to proceed and their performance will be negatively impacted.

Therefore, programs should be designed in such a way that locks are released after the usage is completed using appropriate functions.

Topic 43: Lock Logic consequences

- File locking can generate deadlocks just like deadlocks generated by mutual exclusion locks.
- A read or write may be able to complete its request before encountering a conflicting lock.

- The read or write will return false, and the byte transfer count will be less than the number requested.
- Shared locks if not used properly also generate unexpected errors, as if one process have a shared lock on a file other processes can only ready the file but cannot write into it.

Topic 44: The Registry

The Windows Registry is a hierarchical database that stores low level settings for the Microsoft Windows operating system and for applications that opt to use the registry. The OS kernel, device drivers, services, Security Accounts Manager, and user interfaces all can read and update the registry values.

In other words, the registry or Windows Registry contains different information e.g. settings, options, and other values for programs and hardware installed on the system. For example, when a program is installed, a new key containing settings such as its version is added to the Windows Registry.

Topic 45: Registry Keys

The registry contains two basic elements: keys and values. Registry keys are container objects similar to folders. Registry values are non-container objects similar to files. Keys may contain values and sub-keys. Keys are referenced with a syntax similar to Windows' path names, using backslashes to indicate levels of hierarchy. Keys must have a case insensitive name without backslashes.

The hierarchy of registry keys can only be accessed from a known root key handle (which is anonymous but whose effective value is a constant numeric handle) that is mapped to the content of a registry key pre-loaded by the kernel from a stored "hive", or to the content of a sub-key within another root key, or mapped to a registered service or DLL that provides access to its contained sub-keys and values.

E.g. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows refers to the sub-key "Windows" of the sub-key "Microsoft" of the sub-key "Software" of the HKEY_LOCAL_MACHINE root key.

There are seven predefined root keys, traditionally named according to their constant handles defined in the Win32 API, or by synonymous abbreviations (depending on applications):[4]

- HKEY_LOCAL_MACHINE or HKLM
- HKEY_CURRENT_CONFIG or HKCC
- HKEY_CLASSES_ROOT or HKCR
- HKEY_CURRENT_USER or HKCU
- HKEY_USERS or HKU
- HKEY_PERFORMANCE_DATA (only in Windows NT, but invisible in the Windows Registry Editor)[5]
- HKEY_DYN_DATA (only in Windows 9x, and visible in the Windows Registry Editor)

Topic 46: Key Management

Service Manager stores many settings in the registry. You seldom have to edit the registry yourself, because most of those settings are derived from entries that you make in day-to-day use. However, some changes to settings might occasionally be required. Service Manager stores most registry values in the following locations:

HKEY_CURRENT_USER\Software\Microsoft\System Center<version>\Service Manager\Console
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\System Center<version>

Topic 47: Listing Registry Keys

You can show all items directly within a registry key by using Get-ChildItem. Add the optional Force parameter to display hidden or system items. For example, this command displays the items directly within PowerShell drive HKCU:, which corresponds to the HKEY_CURRENT_USER registry hive:

PowerShell

```
Get-ChildItem -Path HKCU:\ | Select-Object Name
```

You can also specify this registry path by specifying the registry provider's name, followed by ::. The registry provider's full name is Microsoft.PowerShell.Core\Registry, but this can be shortened to just Registry. Any of the following commands will list the contents directly under HKCU:.

PowerShell

```
Get-ChildItem -Path Registry::HKEY_CURRENT_USERGet-ChildItem -Path Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USERGet-ChildItem -Path Registry::HKCUGet-ChildItem -Path Microsoft.PowerShell.Core\Registry::HKCUGet-ChildItem HKCU:
```

Topic 48: Exception and their Handlers

Structured exception handling (SEH) is a Microsoft extension to C to handle certain exceptional code situations, such as hardware faults, gracefully. Although Windows and Microsoft C++ support SEH, we recommend that you use ISO-standard C++ exception handling. It makes your code more portable and flexible. However, to maintain existing code or for particular kinds of programs, you still might have to use SEH.

Topic 49: Try and Except Blocks

Different languages have different implementation for exception handling.

For example Python handles the exception following way:
The try block lets you test a block of code for errors.
The except block lets you handle the error.
The else block lets you execute code when there is no error.
The finally block lets you execute code, regardless of the result of the try- and except blocks.

Topic 50: Filter Expressions

Writing an exception filter in windows

You can handle an exception either by jumping to the level of the exception handler or by continuing execution. Instead of using the exception handler code to handle the exception and falling through, you can use a filter expression to clean up the problem. Then, by returning EXCEPTION_CONTINUE_EXECUTION (-1), you may resume normal flow without clearing the stack.

It's a good idea to use a function call in the filter expression whenever filter needs to do anything complex. Evaluating the expression causes execution of the function, in this case, Eval_Exception.

This handler passes control to another handler unless the exception is an integer or floating-point overflow. If it is, the handler calls a function (ResetVars is only an example, not an API function) to reset some global variables. The __except statement block, which in this example is empty, can never be executed because Eval_Exception never returns EXCEPTION_EXECUTE_HANDLER (1).

Using a function call is a good general-purpose technique for dealing with complex filter expressions. Two other C language features that are useful are:

- The conditional operator
- The comma operator