WaitForSingleObject(): my process waits for the newly created process

CloseHandle() : Handle is a utility that displays information about open handles for any process in the system.  : The **CloseHandle** function closes handles to the following objects:

* Communications device
* Console input
* Console screen buffer
* Event
* File
* File mapping
* Job
* Mailslot
* Mutex
* Named pipe
* Process
* Semaphore
* Socket
* Thread
* Token

The term **console** usually refers to a terminal attached to a minicomputer or mainframe and used to monitor the status of the **system**.

The **SetConsoleMode()** function can be used to set the input and output modes of the console's input and output buffers.

The **ioctl**() function manipulates the underlying device parameters of special files. In particular, many operating characteristics of character special files (e.g. terminals) may be controlled with **ioctl**() requests. The argument *d* must be an open file descriptor.

GetCurrentProcessID() : Retrieves the process identifier of the calling process.

Settimer(): to use SetTimer API to call a function every X minutes. So, i have written this test code.

An operating system “pipe” is an abstraction which implements interprocess communications through an interface which behaves like an open file descriptor.

CreatePipe() : a pipe is a connection between two processes, such that the standard output from one process becomes the standard input of the other process.

The transformation of data from main memory to cache memory is called mapping.

File mapping is the association of a file's contents with a portion of the virtual address space of a process. The system creates a file mapping object (also known as a section object) to maintain this association. A file view is the portion of virtual address space that a process uses to access the file's contents. File mapping allows the process to use both random input and output (I/O) and sequential I/O. It also allows the process to work efficiently with a large data file, such as a database, without having to map the whole file into memory. Multiple processes can also use memory-mapped files to share data.

CreateFileMapping():The first step in mapping a file is to open the file by calling the **[CreateFile](https://msdn.microsoft.com/en-us/library/Aa363858(v=VS.85).aspx)** function. To ensure that other processes cannot write to the portion of the file that is mapped, you should open the file with exclusive access. In addition, the file handle should remain open until the process no longer needs the file mapping object. An easy way to obtain exclusive access is to specify zero in the fdwShareMode parameter of **CreateFile**. The handle returned by **CreateFile** is used by the **[CreateFileMapping](https://docs.microsoft.com/en-us/windows/desktop/api/WinBase/nf-winbase-createfilemappinga)** function to create a file mapping object.

The **[CreateFileMapping](https://docs.microsoft.com/en-us/windows/desktop/api/WinBase/nf-winbase-createfilemappinga)** function returns a handle to the file mapping object. This handle will be used when [creating a file view](https://docs.microsoft.com/en-us/windows/win32/memory/creating-a-file-view) so that you can access the shared memory. When you call **CreateFileMapping**, you specify an object name, the number of bytes to be mapped from the file, and the read/write permission for the mapped memory. The first process that calls **CreateFileMapping** creates the file mapping object. Processes calling **CreateFileMapping**for an existing object receive a handle to the existing object. You can tell whether or not a successful call to **CreateFileMapping** created or opened the file mapping object by calling the **[GetLastError](https://msdn.microsoft.com/en-us/library/ms679360(v=VS.85).aspx)** function. **GetLastError** returns **NO\_ERROR** to the creating process and **ERROR\_ALREADY\_EXISTS** to subsequent processes.

**MapViewOfFile():**Maps a view of a file mapping into the address space of a calling process.

[shmget(): shared memory segment : . In order to create a new message queue, or access an existing queue, the](http://man7.org/linux/man-pages/man2/shmget.2.html)**[shmget()](http://man7.org/linux/man-pages/man2/shmget.2.html)**[system call is used.](http://man7.org/linux/man-pages/man2/shmget.2.html)

**InitializeSecurityDescriptor()**

**security descriptor**

A structure and associated data that contains the security information for a securable object. A security descriptor identifies the object's owner and primary group. It can also contain a DACL that controls access to the object, and a SACL that controls the logging of attempts to access the object.

**SetSecurityDescriptorGroup():** The **SetSecurityDescriptorGroup** function sets the primary group in formation of an absolute-format [security descriptor](https://docs.microsoft.com/windows/desktop/SecGloss/s-gly), replacing any primary group information already present in the security descriptor.

Chmod() : change mode: to change the file mode

The **chmod**() function changes permissions of the specified file.

Unmask(): In computing, **umask** is a command that determines the settings of a mask that controls how file permissions are set for newly created files.

[chown() — Change the owner or group of a file or directory - IBM](https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.2.0/com.ibm.zos.v2r2.bpxbd00/rtcho.htm)