



Console

Programming Windows API

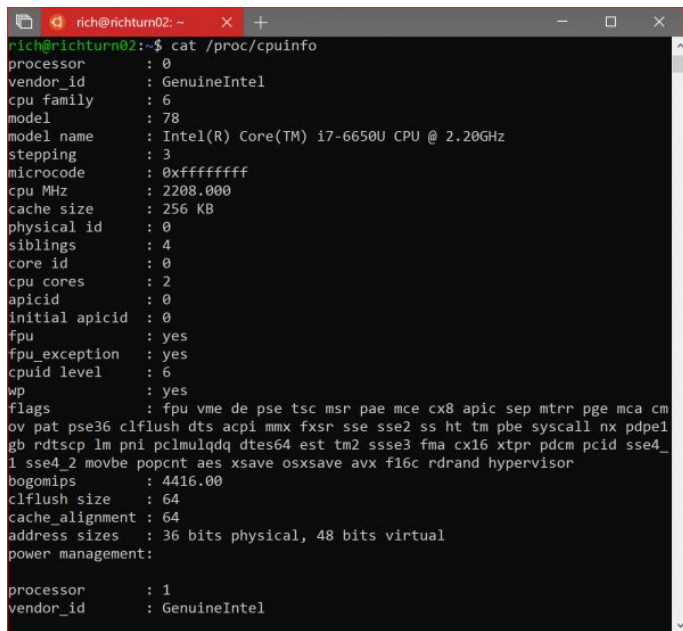
Gintautas Švedas



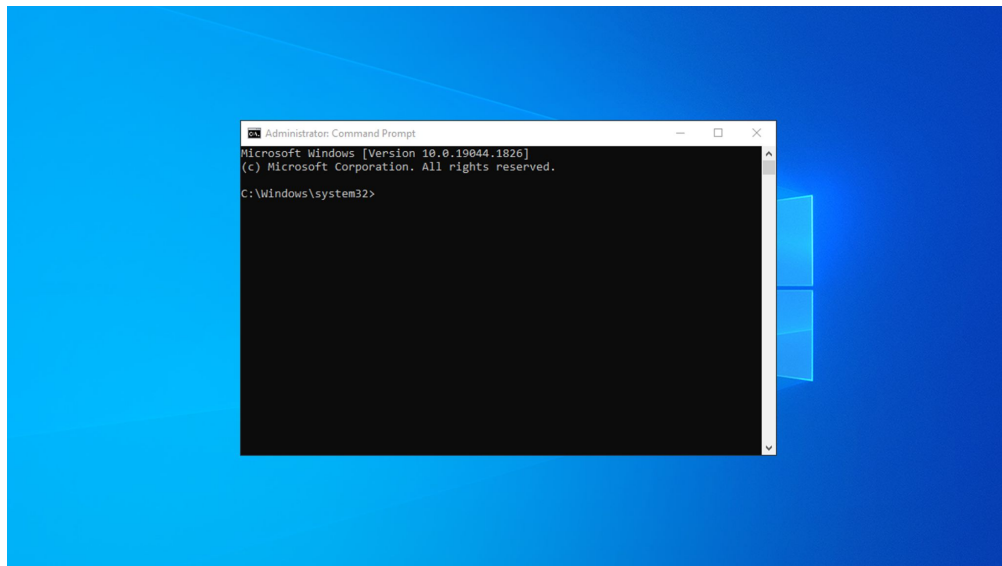
Overview

- What is a console
- Architecture
- Standard Handles
- How does it work, simplified version
- Controller Handlers
- Console Aliases
- Console Security
- Console functions
- Console API structure

What is a console?

A terminal window titled 'rich@richturn02: ~' with a dark background. It shows the output of the command 'cat /proc/cpuinfo'. The output lists various CPU details for an Intel Core i7-6650U, including processor ID, vendor, family, model, name, stepping, microcode, frequency, cache size, physical ID, siblings, core ID, cores, apicid, initial apicid, FPU, exceptions, CPUID level, WP, flags, and power management settings. The window has standard Linux terminal window controls at the top.

```
rich@richturn02: ~  
rich@richturn02:~$ cat /proc/cpuinfo  
processor       : 0  
vendor_id      : GenuineIntel  
cpu family     : 6  
model          : 78  
model name     : Intel(R) Core(TM) i7-6650U CPU @ 2.20GHz  
stepping       : 3  
microcode      : 0xffffffff  
cpu MHz        : 2208.000  
cache size     : 256 KB  
physical id    : 0  
siblings       : 4  
core id        : 0  
cpu cores      : 2  
apicid         : 0  
initial apicid : 0  
fpu            : yes  
fpu_exception  : yes  
cpuid level    : 6  
wp             : yes  
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cm  
ov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1  
gb rdtscp lm pni pclmulqdq dtes64 est tm2 ssse3 fma cx16 xtpr pdcm pcid sse4_  
l sse4_2 movbe popcnt aes xsave osxsave avx f16c rdrand hypervisor  
bogomips      : 4416.00  
clflush size   : 64  
cache_alignme  : 64  
address sizes  : 36 bits physical, 48 bits virtual  
power management:  
  
processor       : 1  
vendor_id      : GenuineIntel
```





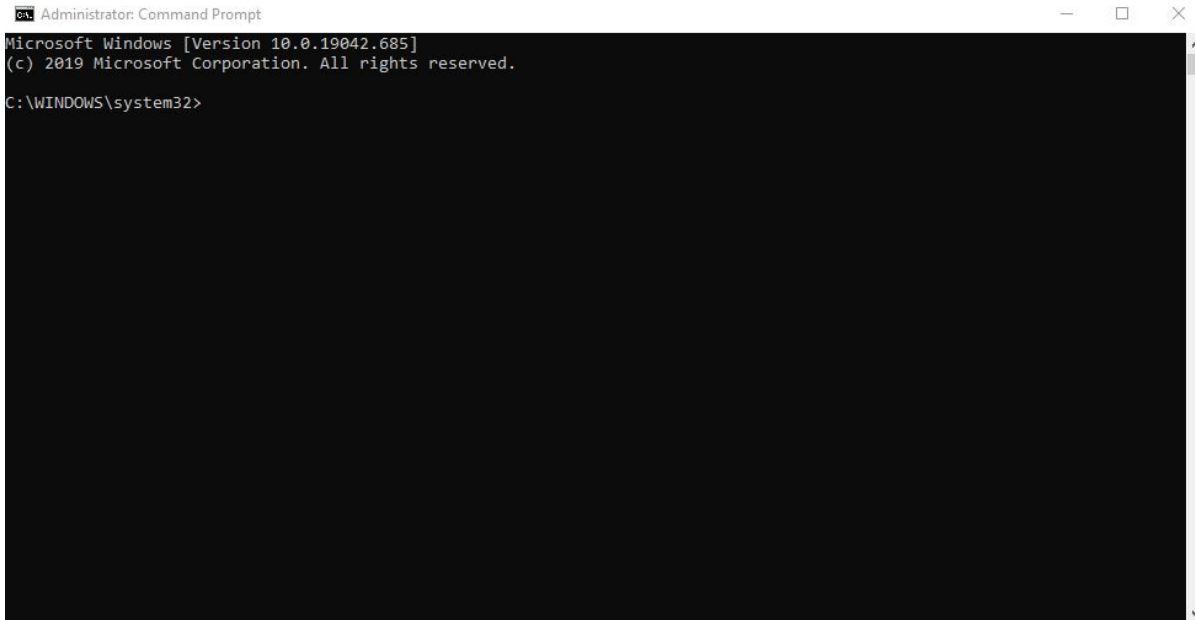
Architecture

Command-Line Communication





How does it look?



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.685]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>
```



Standard handles

- STDIN
- STDOUT
- STDERR



How does it work, simplified version

- [Optionally] Read data from standard input (stdin).
- Do "work".
- [Optionally] Write data to standard output (stdout) or standard error (stderr).



Controller Handlers

- CTRL+C
- CTRL+BREAK
- CTRL+CLOSE



Console Aliases


- Console aliases are used to map source strings to target strings.

EXAMPLE: a console alias that maps "test" to "cd \custom path\test".

`doskey test=cd \a_very_long_path\test`

-

C

 Copy

```
AddConsoleAlias( TEXT("test"),  
                  TEXT("cd \\<a_very_long_path>\\test"),  
                  TEXT("cmd.exe"));
```



Console Security

- The Windows security model enables you to control access to console input buffers and console screen buffers
- You can specify a security descriptor for the console input and console screen buffers when you call the `CreateFile` or `CreateConsoleScreenBuffer` function.
- Access rights:

Value	Meaning
<code>GENERIC_READ (0x80000000L)</code>	Requests read access to the console screen buffer, enabling the process to read data from the buffer.
<code>GENERIC_WRITE (0x40000000L)</code>	Requests write access to the console screen buffer, enabling the process to write data to the buffer.



Console functions



Console functions (A - F)

Function	Description
AddConsoleAlias	Defines a console alias for the specified executable.
AllocConsole	Allocates a new console for the calling process.
AttachConsole	Attaches the calling process to the console of the specified process.
ClosePseudoConsole	Closes a pseudoconsole from the given handle.
CreatePseudoConsole	Allocates a new pseudoconsole for the calling process.
CreateConsoleScreenBuffer	Creates a console screen buffer.
FillConsoleOutputAttribute	Sets the text and background color attributes for a specified number of character cells.
FillConsoleOutputCharacter	Writes a character to the console screen buffer a specified number of times.
FlushConsoleInputBuffer	Flushes the console input buffer.
FreeConsole	Detaches the calling process from its console.



Console functions (G - G)

<code>GenerateConsoleCtrlEvent</code>	Sends a specified signal to a console process group that shares the console associated with the calling process.
<code>GetConsoleAlias</code>	Retrieves the specified alias for the specified executable.
<code>GetConsoleAliases</code>	Retrieves all defined console aliases for the specified executable.
<code>GetConsoleAliasesLength</code>	Returns the size, in bytes, of the buffer needed to store all of the console aliases for the specified executable.
<code>GetConsoleAliasExes</code>	Retrieves the names of all executables with console aliases defined.
<code>GetConsoleAliasExesLength</code>	Returns the size, in bytes, of the buffer needed to store the names of all executables that have console aliases defined.
<code>GetConsoleCP</code>	Retrieves the input code page used by the console associated with the calling process.
<code>GetConsoleCursorInfo</code>	Retrieves information about the size and visibility of the cursor for the specified console screen buffer.
<code>GetConsoleDisplayMode</code>	Retrieves the display mode of the current console.
<code>GetConsoleFontSize</code>	Retrieves the size of the font used by the specified console screen buffer.



Console functions (G - G)

<code>GetConsoleMode</code>	Retrieves the current input mode of a console's input buffer or the current output mode of a console screen buffer.
<code>GetConsoleOriginalTitle</code>	Retrieves the original title for the current console window.
<code>GetConsoleOutputCP</code>	Retrieves the output code page used by the console associated with the calling process.
<code>GetConsoleProcessList</code>	Retrieves a list of the processes attached to the current console.
<code>GetConsoleScreenBufferInfo</code>	Retrieves information about the specified console screen buffer.
<code>GetConsoleScreenBufferInfoEx</code>	Retrieves extended information about the specified console screen buffer.
<code>GetConsoleSelectionInfo</code>	Retrieves information about the current console selection.
<code>GetConsoleTitle</code>	Retrieves the title for the current console window.
<code>GetConsoleWindow</code>	Retrieves the window handle used by the console associated with the calling process.



Console functions (G - R)

GetCurrentConsoleFont	Retrieves information about the current console font.
GetCurrentConsoleFontEx	Retrieves extended information about the current console font.
GetLargestConsoleWindowSize	Retrieves the size of the largest possible console window.
GetNumberOfConsoleInputEvents	Retrieves the number of unread input records in the console's input buffer.
GetNumberOfConsoleMouseButtons	Retrieves the number of buttons on the mouse used by the current console.
GetStdHandle	Retrieves a handle for the standard input, standard output, or standard error device.
HandlerRoutine	An application-defined function used with the SetConsoleCtrlHandler function.
PeekConsoleInput	Reads data from the specified console input buffer without removing it from the buffer.
ReadConsole	Reads character input from the console input buffer and removes it from the buffer.
ReadConsoleInput	Reads data from a console input buffer and removes it from the buffer.



Console functions (R - S)

ReadConsoleInputEx	Reads data from a console input buffer and removes it from the buffer, with configurable behavior.
ReadConsoleOutput	Reads character and color attribute data from a rectangular block of character cells in a console screen buffer.
ReadConsoleOutputAttribute	Copies a specified number of foreground and background color attributes from consecutive cells of a console screen buffer.
ReadConsoleOutputCharacter	Copies a number of characters from consecutive cells of a console screen buffer.
ResizePseudoConsole	Resizes the internal buffers for a pseudoconsole to the given size.
ScrollConsoleScreenBuffer	Moves a block of data in a screen buffer.
SetConsoleActiveScreenBuffer	Sets the specified screen buffer to be the currently displayed console screen buffer.
SetConsoleCP	Sets the input code page used by the console associated with the calling process.
SetConsoleCtrlHandler	Adds or removes an application-defined HandlerRoutine from the list of handler functions for the calling process.
SetConsoleCursorInfo	Sets the size and visibility of the cursor for the specified console screen buffer.



Console functions (S - S)

SetConsoleCursorPosition	Sets the cursor position in the specified console screen buffer.
SetConsoleDisplayMode	Sets the display mode of the specified console screen buffer.
SetConsoleHistoryInfo	Sets the history settings for the calling process's console.
SetConsoleMode	Sets the input mode of a console's input buffer or the output mode of a console screen buffer.
SetConsoleOutputCP	Sets the output code page used by the console associated with the calling process.
SetConsoleScreenBufferInfoEx	Sets extended information about the specified console screen buffer.
SetConsoleScreenBufferSize	Changes the size of the specified console screen buffer.
SetConsoleTextAttribute	Sets the foreground (text) and background color attributes of characters written to the console screen buffer.
SetConsoleTitle	Sets the title for the current console window.
SetConsoleWindowInfo	Sets the current size and position of a console screen buffer's window.





Console functions (S - S)

<code>SetCurrentConsoleFontEx</code>	Sets extended information about the current console font.
<code>SetStdHandle</code>	Sets the handle for the standard input, standard output, or standard error device.
<code>WriteConsole</code>	Writes a character string to a console screen buffer beginning at the current cursor location.
<code>WriteConsoleInput</code>	Writes data directly to the console input buffer.
<code>WriteConsoleOutput</code>	Writes character and color attribute data to a specified rectangular block of character cells in a console screen buffer.
<code>WriteConsoleOutputAttribute</code>	Copies a number of foreground and background color attributes to consecutive cells of a console screen buffer.
<code>WriteConsoleOutputCharacter</code>	Copies a number of characters to consecutive cells of a console screen buffer.



Console API structure

Console API structure

- CHAR_INFO
- CONSOLE_CURSOR_INFO 
- CONSOLE_FONT_INFO
- CONSOLE_FONT_INFOEX
- CONSOLE_HISTORY_INFO
- CONSOLE_READCONSOLE_CONTROL
- CONSOLE_SCREEN_BUFFER_INFO
- CONSOLE_SCREEN_BUFFER_INFOEX
- CONSOLE_SELECTION_INFO
- COORD
- FOCUS_EVENT_RECORD
- INPUT_RECORD
- KEY_EVENT_RECORD 
- MENU_EVENT_RECORD
- MOUSE_EVENT_RECORD

```
typedef struct _CONSOLE_CURSOR_INFO
{
    DWORD dwSize;
    BOOL bVisible;
} CONSOLE_CURSOR_INFO,
*PCONSOLE_CURSOR_INFO;
```

```
typedef struct _KEY_EVENT_RECORD {
    BOOL bKeyDown;
    WORD wRepeatCount;
    WORD wVirtualKeyCode;
    WORD wVirtualScanCode;
    union {
        WCHAR UnicodeChar;
        CHAR AsciiChar;
    } uChar;
    DWORD dwControlKeyState;
} KEY_EVE
```



Thank you!