

# How to check CPU info on Linux

**Author :** Dan Nanni

**Categories :** [Hardware](#)

**Tagged as :** [bios](#), [cpunuma](#)

**Question:** I would like to know detailed information about the CPU processor of my computer. What are the available methods to check CPU information on Linux?

Depending on your need, there are various pieces of information you may need to know about the CPU processor(s) of your computer, such as CPU vendor name, model name, clock speed, number of sockets/cores, L1/L2/L3 cache configuration, available processor capabilities (e.g., hardware virtualization, AES, MMX, SSE), and so on. In Linux, there are many command line or GUI-based tools that are used to show detailed information about your CPU hardware.

## 1. /proc/cpuinfo

The simplest method is to check /proc/cpuinfo. This virtual file shows the configuration of available CPU hardware.

```
$ more /proc/cpuinfo
```

```
$ more /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 58
model name     : Intel(R) Core(TM) i7-3612QM CPU @ 2.10GHz
stepping       : 9
microcode      : 0x12
cpu MHz        : 1200.000
cache size     : 6144 KB
physical id    : 0
siblings       : 8
core id        : 0
cpu cores      : 4
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 13
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx rdtscp lm c
onstant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf
eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm p
cid sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx f16c rdrand lah
f_lm ida arat epb xsaveopt pln pts dtherm tpr_shadow vnmi flexpriority ept vpid
fsgsbase smep erms
bogomips       : 4190.56
clflush size   : 64
cache_alignmen : 64
address sizes   : 36 bits physical, 48 bits virtual
power managemen:
```

By inspecting this file, you can [identify](#) the number of physical processors, the number of cores per CPU, available CPU flags, and a number of other things.

## 2. cpufreq-info

The `cpufreq-info` command (which is part of `cpufrequtils` package) collects and reports CPU frequency information from the kernel/hardware. The command shows the hardware frequency that the CPU currently runs at, as well as the minimum/maximum CPU frequency allowed, CPUfreq policy/statistics, and so on. To check up on CPU #0:

```
$ cpufreq-info -c 0
```

```
$ cpufreq-info -c 0
cpufrequtils 008: cpufreq-info (C) Dominik Brodowski 2004-2009
Report errors and bugs to cpufreq@vger.kernel.org, please.
analyzing CPU 0:
  driver: acpi-cpufreq
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 10.0 us.
  hardware limits: 1.20 GHz - 2.10 GHz
  available frequency steps: 2.10 GHz, 2.10 GHz, 2.00 GHz, 1.90 GHz, 1.80 GHz, 1.70 GHz, 1.60 GHz, 1.50 GHz, 1.40 GHz, 1.30 GHz, 1.20 GHz
  available cpufreq governors: conservative, ondemand, userspace, powersave, performance
  current policy: frequency should be within 1.20 GHz and 2.10 GHz.
                   The governor "ondemand" may decide which speed to use
                   within this range.
  current CPU frequency is 1.20 GHz.
  cpufreq stats: 2.10 GHz:13.92%, 2.10 GHz:0.00%, 2.00 GHz:0.73%, 1.90 GHz:0.55%, 1.80 GHz:0.33%, 1.70 GHz:0.41%, 1.60 GHz:0.39%, 1.50 GHz:0.37%, 1.40 GHz:0.42%, 1.30 GHz:0.23%, 1.20 GHz:82.65% (2127)
```

### 3. cpuid

The `cpuid` command-line utility is a dedicated CPU information tool that displays verbose information about CPU hardware by using [CPUID functions](#). Reported information includes processor type/family, CPU extensions, cache/TLB configuration, power management features, etc.

```
$ cpuid
```

```
$ cpuid | more
CPU 0:
  vendor_id = "GenuineIntel"
  version information (1/eax):
    processor type = primary processor (0)
    family        = Intel Pentium Pro/II/III/Celeron/Core/Core 2/Atom, AMD A
thlon/Duron, Cyrix M2, VIA C3 (6)
    model         = 0xa (10)
    stepping id    = 0x9 (9)
    extended family = 0x0 (0)
    extended model = 0x3 (3)
    (simple synth) = Intel Core i3-3000 (Ivy Bridge L1) / i5-3000 (Ivy Bridge
E1/N0/L1) / i7-3000 (Ivy Bridge E1) / Mobile Core i3-3000 (Ivy Bridge L1) / i5-
3000 (Ivy Bridge L1) / Mobile Core i7-3000 (Ivy Bridge E1/L1) / Xeon E3-1200 v2
(Ivy Bridge E1/N0/L1) / Pentium G1600/G2000/G2100 (Ivy Bridge P0) / Pentium 900/
1000/2000/2100 (P0), 22nm
  miscellaneous (1/ebx):
    process local APIC physical ID = 0x0 (0)
    cpu count                      = 0x10 (16)
    CLFLUSH line size              = 0x8 (8)
    brand index                    = 0x0 (0)
  brand id = 0x00 (0): unknown
  feature information (1/edx):
    x87 FPU on chip                = true
    virtual-8086 mode enhancement = true
    debugging extensions           = true
    page size extensions           = true
    time stamp counter             = true
    RDMSR and WRMSR support        = true
    physical address extensions    = true
```

## 4. dmidecode

The `dmidecode` command collects detailed information about system hardware directly from DMI data of the BIOS. Reported CPU information includes CPU vendor, version, CPU flags, maximum/current clock speed, (enabled) core count, L1/L2/L3 cache configuration, and so on.

```
$ sudo dmidecode
```



```
Handle 0x0004, DMI type 4, 42 bytes
Processor Information
    Socket Designation: U3E1
    Type: Central Processor
    Family: Core i7
    Manufacturer: Intel(R) Corporation
    ID: A9 06 03 00 FF FB EB BF
    Signature: Type 0, Family 6, Model 58, Stepping 9
    Flags:
        FPU (Floating-point unit on-chip)
        VME (Virtual mode extension)
        DE (Debugging extension)
        PSE (Page size extension)
        TSC (Time stamp counter)
        MSR (Model specific registers)
        PAE (Physical address extension)
        MCE (Machine check exception)
        CX8 (CMPXCHG8 instruction supported)
        APIC (On-chip APIC hardware supported)
        SEP (Fast system call)
        MTRR (Memory type range registers)
        PGE (Page global enable)
        MCA (Machine check architecture)
        CMOV (Conditional move instruction supported)
        PAT (Page attribute table)
        PSE-36 (36-bit page size extension)
        CLFSH (CLFLUSH instruction supported)
```

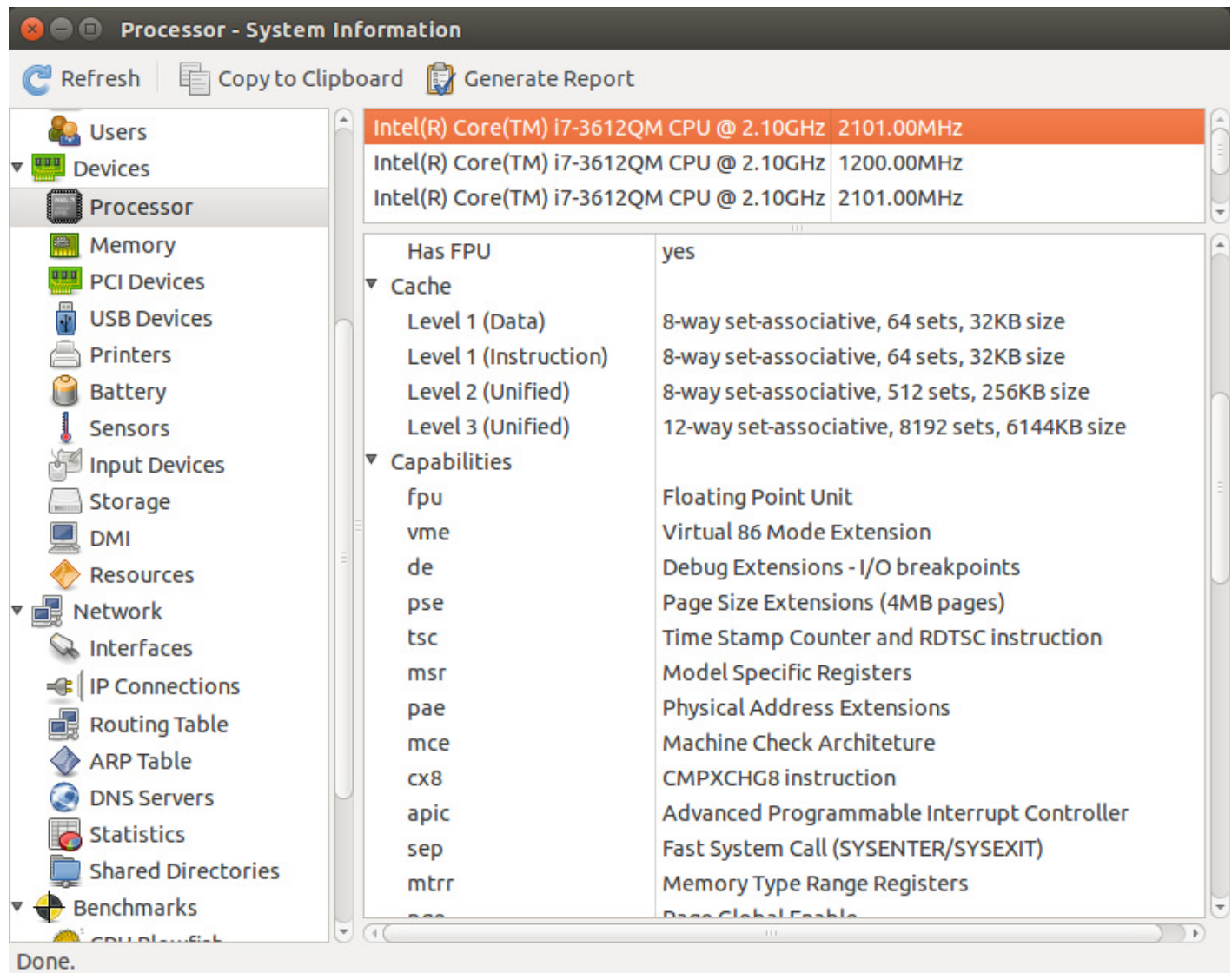
## 5. hardinfo

The `hardinfo` is a GUI-based system information tool which can give you an easy-to-understand summary of your CPU hardware, as well as other hardware components of your system.

```
$ hardinfo
```

## Ask Xmodulo

Find answers to commonly asked Linux questions  
<http://ask.xmodulo.com>



## 6. i7z

`i7z` is a real-time CPU reporting tool dedicated to Intel Core i3, i5 and i7 CPUs. It can display various per-core information in real time, such as Turbo Boost states, CPU frequencies, CPU power states, temperature measurements, and so on. `i7z` runs in either `ncurses`-based console mode or QT based GUI.

```
$ sudo i7z
```

## Ask Xmodulo

Find answers to commonly asked Linux questions

<http://ask.xmodulo.com>

```
Cpu speed from cpufreq 2095.00Mhz
cpufreq might be wrong if cpufreq is enabled. To guess correctly try estimating via tsc
Linux's inbuilt cpu khz code emulated now
True Frequency (without accounting Turbo) 2095 MHz
CPU Multiplier 21x || Bus clock frequency (BCLK) 99.76 MHz

Socket [0] - [physical cores=4, logical cores=8, max online cores ever=4]
TURBO ENABLED on 4 Cores, Hyper Threading ON
Max Frequency without considering Turbo 2194.76 MHz (99.76 x [22])
Max TURBO Multiplier (if Enabled) with 1/2/3/4 Cores is 31x/30x/28x/28x
Real Current Frequency 1514.79 MHz [99.76 x 15.18] (Max of below)

```

Core [core-id]	:Actual Freq (Mult.)	C0%	Halt(C1)%	C3 %	C6 %	C7 %	Temp
Core 1 [0]:	1245.80 (12.49x)	4.41	5.46	0	0	91.9	44
Core 2 [2]:	1217.46 (12.20x)	1.42	3.15	1	0	95	45
Core 3 [4]:	1404.82 (14.08x)	2.98	1.47	1	0	95.5	47
Core 4 [6]:	1514.79 (15.18x)	4.21	2.34	1	0	93.6	47

```

C0 = Processor running without halting
C1 = Processor running with halts (States >C0 are power saver)
C3 = Cores running with PLL turned off and core cache turned off
C6 = Everything in C3 + core state saved to last level cache
Above values in table are in percentage over the last 1 sec
[core-id] refers to core-id number in /proc/cpuinfo
'Garbage Values' message printed when garbage values are read
Ctrl+C to exit

```

## 7. inxi

`inxi` is a bash script written to gather system information in a human-friendly format. It shows a quick summary of CPU information including CPU model, cache size, clock speed, and supported CPU capabilities.

```
$ inxi -C
```

```
$ inxi -C
CPU:      Quad core Intel Core i7-3612QM CPU (-HT-MCP-) cache: 6144 KB flags: (
lm nx sse sse2 sse3 sse4_1 sse4_2 ssse3 vmx)
Clock Speeds: 1: 1200.00 MHz 2: 1200.00 MHz 3: 1200.00 MHz 4: 2101.00
MHz 5: 1200.00 MHz 6: 1200.00 MHz 7: 1200.00 MHz 8: 2101.00 MHz
$
$
```

## 8. likwid-topology

[likwid](#) (Like I Knew What I'm Doing) is a collection of command-line tools to measure, configure and display hardware related properties. Among them is `likwid-topology` which shows CPU hardware



(thread/cache/NUMA) topology information. It can also identify processor families (e.g., Intel Core 2, AMD Shanghai).

```
$ likwid-topology
-----
CPU type:      Intel Core IvyBridge processor
*****
Hardware Thread Topology
*****
Sockets:      1
Cores per socket:  4
Threads per core:  2
-----
HWThread      Thread      Core      Socket
0             0           0          0
1             1           0          0
2             0           1          0
3             1           1          0
4             0           2          0
5             1           2          0
6             0           3          0
7             1           3          0
-----
Socket 0: ( 0 1 2 3 4 5 6 7 )
-----

*****
Cache Topology
*****
Level:  1
Size:   32 kB
Cache groups:  ( 0 1 ) ( 2 3 ) ( 4 5 ) ( 6 7 )
-----
Level:  2
Size:  256 kB
Cache groups:  ( 0 1 ) ( 2 3 ) ( 4 5 ) ( 6 7 )
-----
Level:  3
```

## 9. lscpu

The `lscpu` command summarizes `/etc/cpuinfo` content in a more user-friendly format, e.g., the number of (online/offline) CPUs, cores, sockets, NUMA nodes.

```
$ lscpu
```



```
$ lscpu
Architecture:            x86_64
CPU op-mode(s):          32-bit, 64-bit
Byte Order:               Little Endian
CPU(s):                   8
On-line CPU(s) list:     0-7
Thread(s) per core:       2
Core(s) per socket:      4
Socket(s):                1
NUMA node(s):            1
Vendor ID:                GenuineIntel
CPU family:               6
Model:                   58
Stepping:                 9
CPU MHz:                  1200.000
BogoMIPS:                 4190.56
Virtualization:           VT-x
L1d cache:                32K
L1i cache:                32K
L2 cache:                 256K
L3 cache:                 6144K
NUMA node0 CPU(s):       0-7
```

## 10. Lshw

The `lshw` command is a comprehensive hardware query tool. Unlike other tools, `lshw` requires root privilege because it queries DMI information in system BIOS. It can report the total number of cores and enabled cores, but miss out on information such as L1/L2/L3 cache configuration. The GTK version `lshw-gtk` is also available.

```
$ sudo lshw -class processor
```

```
$ sudo lshw -class processor
*-cpu
  description: CPU
  product: Intel(R) Core(TM) i7-3612QM CPU @ 2.10GHz
  vendor: Intel Corp.
  physical id: 4
  bus info: cpu@0
  version: Intel(R) Core(TM) i7-3612QM CPU @ 2.10GHz
  serial: To Be Filled By O.E.M.
  slot: U3E1
  size: 1300MHz
  capacity: 4GHz
  width: 64 bits
  clock: 100MHz
  capabilities: x86-64 fpu fpu_exception wp vme de pse tsc msr pae mce cx8
apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm
pbe syscall nx rdtscp constant_tsc arch_perfmon pebs bts rep_good nopl xtopolog
y nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx est tm
2 ssse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes x
save avx f16c rdrand lahf_lm ida arat epb xsaveopt pln pts dtherm tpr_shadow vnm
i flexpriority ept vpid fsgsbase smep erms cpufreq
  configuration: cores=4 enabledcores=4 threads=8
```

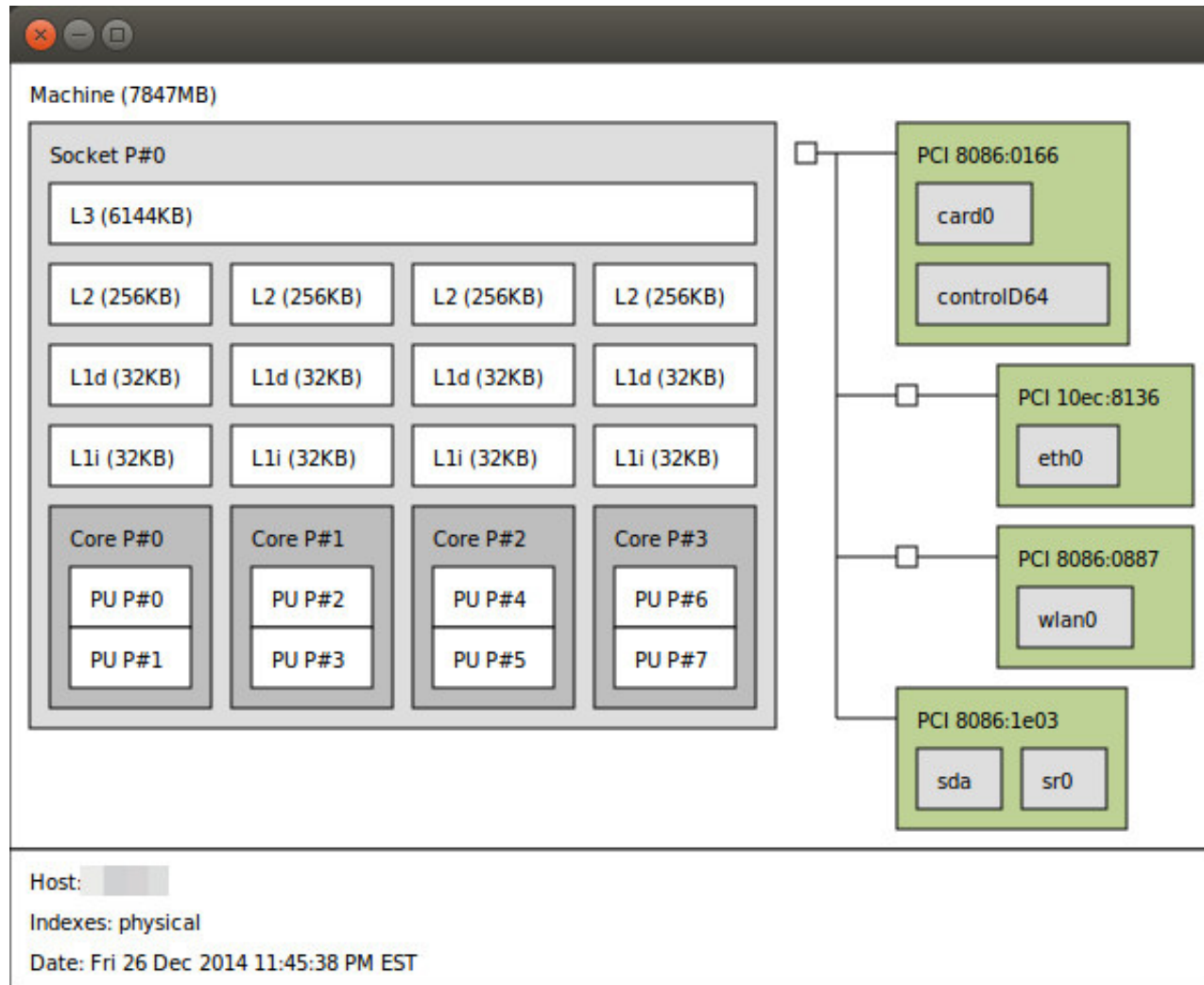
## 11. lstopo

The `lstopo` command (contained in [hwloc](#) package) visualizes the topology of the system which is composed of CPUs, cache, memory and I/O devices. This command is useful to identify the processor architecture and NUMA topology of the system.

```
$ lstopo
```

## Ask Xmodulo

Find answers to commonly asked Linux questions  
<http://ask.xmodulo.com>



## 12. numactl

Originally developed to set the NUMA scheduling and memory placement policy of Linux processes, the `numactl` command can also show information about NUMA topology of the CPU hardware from the command line.

```
$ numactl --hardware
```

```
$ numactl --hardware
available: 1 nodes (0)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 7847 MB
node 0 free: 1199 MB
node distances:
node    0
  0:    10
$
```

### 13. x86info

`x86info` is a command-line tool for showing x86-based CPU information. Reported information includes CPU model, number of threads/cores, clock speed, TLB cache configuration, supported feature flags, etc.

```
$ x86info --all
```

```
$ x86info --all
x86info v1.30. Dave Jones 2001-2011
Feedback to <davej@redhat.com>.

Found 8 identical CPUs
Extended Family: 0 Extended Model: 3 Family: 6 Model: 58 Stepping: 9
Type: 0 (Original OEM)
CPU Model (x86info's best guess): Unknown model.
Processor name string (BIOS programmed): Intel(R) Core(TM) i7-3612QM CPU @ 2.10GHz

eax in: 0x00000000, eax = 0000000d ebx = 756e6547 ecx = 6c65746e edx = 49656e69
eax in: 0x00000001, eax = 000306a9 ebx = 00100800 ecx = 7fbae3bf edx = bfebfbff
eax in: 0x00000002, eax = 76035a01 ebx = 00f0b2ff ecx = 00000000 edx = 00ca0000
eax in: 0x00000003, eax = 00000000 ebx = 00000000 ecx = 00000000 edx = 00000000
eax in: 0x00000004, eax = 1c004121 ebx = 01c0003f ecx = 0000003f edx = 00000000
eax in: 0x00000005, eax = 00000040 ebx = 00000040 ecx = 00000003 edx = 00021120
eax in: 0x00000006, eax = 00000077 ebx = 00000002 ecx = 00000009 edx = 00000000
eax in: 0x00000007, eax = 00000000 ebx = 00000000 ecx = 00000000 edx = 00000000
eax in: 0x00000008, eax = 00000000 ebx = 00000000 ecx = 00000000 edx = 00000000
eax in: 0x00000009, eax = 00000000 ebx = 00000000 ecx = 00000000 edx = 00000000
eax in: 0x0000000a, eax = 07300403 ebx = 00000000 ecx = 00000000 edx = 00000603
```