

Como obter o CPUID no Windows

O CPUID é um código único de identificação que cada processador possui. Ele é utilizado para identificar características específicas do processador, como o modelo, a velocidade de clock, a arquitetura, entre outros dados. Neste artigo, vamos apresentar exemplos de script em linguagem de programação para obter o CPUID no sistema operacional Windows.

Exemplos:

1. Exemplo em C++:

```
#include <iostream>
#include <intrin.h>

int main() { int CPUInfo[4] = { 0, 0, 0, 0 }; __cpuid(CPUInfo, 0); int nIds = CPUInfo[0];

char CPUBrandString[0x40];
memset(CPUBrandString, 0, sizeof(CPUBrandString));

for (int i = 0x80000000; i <= nIds; ++i) {
    __cpuid(CPUInfo, i);
    if (i == 0x80000002)
        memcpy(CPUBrandString, CPUInfo, sizeof(CPUInfo));
    else if (i == 0x80000003)
        memcpy(CPUBrandString + 16, CPUInfo, sizeof(CPUInfo));
    else if (i == 0x80000004)
        memcpy(CPUBrandString + 32, CPUInfo, sizeof(CPUInfo));
}

std::cout << "CPUID: " << CPUBrandString << std::endl;

return 0;

}
```

2. Exemplo em Python:

```
```python
import subprocess

def get_cpu_id():
 output = subprocess.check_output("wmic cpu get ProcessorID", shell=True
```

e)

```
cpu_id = output.decode().strip().split("\n")[1]
return cpu_id
```

```
if __name__ == "__main__":
 cpu_id = get_cpu_id()
 print("CPUID:", cpu_id)
```

**Conclusão:** Obter o CPUID no Windows é uma tarefa relativamente simples utilizando algumas funções específicas disponíveis em linguagens de programação. Nos exemplos apresentados, mostramos como obter o CPUID em C++ e Python. É importante ressaltar que o CPUID pode fornecer informações úteis para diferentes finalidades, como diagnóstico de problemas, otimização de desempenho e compatibilidade de software. Portanto, conhecer e utilizar o CPUID pode ser benéfico para desenvolvedores e entusiastas de tecnologia.