Optimalizált vs Debug build

1. **Felhasznált eszközök**

A méréseket egy Lenovo ThinkPad P50 laptopon végeztem, Windows 10 OS alatt, Intel Core i7-6820HQ processzoron. Továbba, az „Intel(R) Power Gadget 3.6” nevű alkalmazást, a kód futtatásához, illetve optimalizálásához pedig a Micorsoft Visual Studio fejlesztő környezetét használtam.

1. **A futtatott kód**

A méréseket a hátizsák probléma futtatása közben végeztem. Ennek az implementációja látható lejjebb. Elég nagy bemenet esetén (pl. 500 szám pár) elég sokáig fut, hogy a mérés lehetséges legyen. Mindenképpen egy NP-nehéz problémával akartam próbálkozni, ami ráadásul relatíve sok helyet foglal el, hogy mindkét féle optimalizációnak látható legyen a hatása.

|  |
| --- |
| #include <iostream>  #include <cstdlib>  int max(int a, int b) { return (a > b) ? a : b; }  int knapSack(int capacity, int weights[], int values[], int n)  {  if (n == 0 || capacity == 0)  return 0;  if (weights[n - 1] > capacity)  return knapSack(capacity, weights, values, n - 1);  return max(  values[n - 1] + knapSack(capacity - weights[n - 1], weights, values, n - 1),  knapSack(capacity, weights, values, n - 1)  );  }  void fillArrays(int values[], int weights[])  {  for (int i = 0; i < sizeof(values) / sizeof(int); i++)  values[i] = rand() % 100 + 1;  for (int i = 0; i < sizeof(weights) / sizeof(int); i++)  values[i] = rand() % 50 + 1;  }  int main()  {  const int numValues = 500;  srand((unsigned)time(NULL));  int values[numValues];  int weights[numValues];  fillArrays(values, weights);  int capacity = 3 \* rand() % 150 + 1;  int n = sizeof(values) / sizeof(values[0]);  std::cout << knapSack(capacity, weights, values, n) << '\n';  } |

1. **A mérések**

A kód futtatása előtt 11.80W volt a teljesítmény, illetve 40°C -os volt (mindez akkor, amikor a Power Gadget, egy Word dokumentum, illetve a Visual Studio volt megnyitva).

Először Debug módban mértem ötször egymás után. A Start Log gombra kattintva elindult a loggolás. Miután importáltam Excelbe az eredményeket a következő átlagokat kaptam:

Eltelt idő (sec) = 20,18

Elhasznált energia (J) = 550,12

Átlagos teljesítmény (W) = 27,26

Átlagos hőmérséklet (°C) = 65,26

Ez után O1 optimalizációval próbálkoztam, ami a minimális méretre törekszik. Ebben az esetben is 5 mérést végeztem, ennek az eredménye:

Eltelt idő (sec) = 18,71

Elhasznált energia (J) = 483,31

Átlagos teljesítmény (W) = 25,88

Átlagos hőmérséklet (°C) = 65,61

O2 optimalizáció (sebességre optimalizál) esetén pedig:

Eltelt idő (sec) = 17,63

Elhasznált energia (J) = 465,49

Átlagos teljesítmény (W) = 26,41

Átlagos hőmérséklet (°C) = 65,46

1. **Összesített táblázat**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Debug** | **O1 optimalizáció** | **O2 optimalizáció** |
| **Eltelt idő (sec)** | 20,18 | 18,71 | 17,63 |
| **Elhasznált energia (J)** | 550,12 | 483,31 | 465,49 |
| **Átlagos teljesítmény (W)** | 27,26 | 25,88 | 26,41 |
| **Átlagos hőmérséklet (°C)** | 65,26 | 65,61 | 65,46 |

Ami meglepetés volt, hogy a Debug mód mennyivel rosszabb energiafelhasználás szempontjából. A sebességgel kapcsolatos elképzeléseimet igazolták a mérések, valóban az O2 eredményezi a leggyorsabb, a Debug mód pedig a leglassabb lefutást.

1. **Mérési hibák okai**

Több oka is van annak, hogy a mérések nem tökéletesek. Egyrészt nem tudtam egyszerre indítani a mérést a kóddal, illetve leállítani sem tudtam a mérést akkor, amikor a kód lefutott. Másrészt, random értékekkel töltöttem fel a tömböket, ami kis mértékben befolyásolhatja az eredményeket (bár az egyes esetekben nem volt mérvadó különbség 1-1 futás között). Továbbá, biztosan létezik olyan program kód, ami sokkal jobban teszteli az energia használatát a processzornak, bár úgy gondolom, hogy a célnak a hátizsák probléma is megfelel.

1. **Megjegyzés**

A beírt értékek két tizedesre vannak kerekítve, a Power Gadget ennél valamivel nagyobb pontossággal mért.

Az összes log fájl, a forráskód és a dokumentáció megtalálható ebben a [GitHub repository](https://github.com/pasztork/imsc-1-hazi-feladat)-ban.