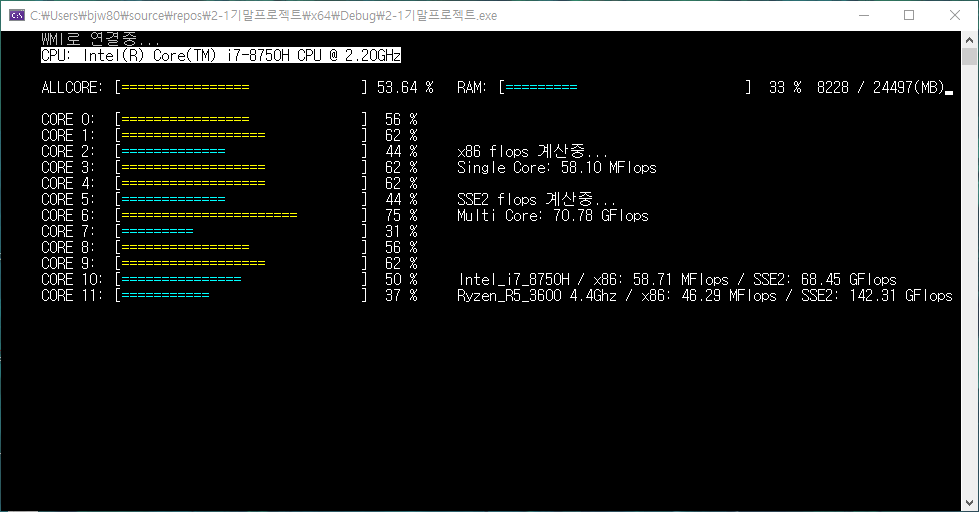
**Instruction**

2019253066 컴퓨터정보통신공학부 복정우

**=프로젝트 간단 설명=**

내 컴퓨터/노트북의 CPU 성능과 점유율, 클럭, 온도 등을 실시간으로 체크하고 비교할 수 있는 프로그램입니다.



**=프로젝트의 배경=**

단순히 강의 과제로써, 만들면 다시 쓰지 않을 프로그램 보다는 실용적인 프로그램을 만들고 싶어 고민하던 중, 제가 사용하던 노트북이나 컴퓨터가 얼마나 일을 하고 있는지 궁금하게 되었습니다. 그리하여 컴퓨터의 각종 정보를 실시간으로 갱신하여 콘솔창에 보기 쉽게 보여주는 프로그램을 구상하게 되었습니다.

**=프로그램 구성=**

Base 클래스는 다른 클래스들의 기반 클래스 입니다.

구성 함수

private:

float Usage; //\_fCPUv; = float allcore; = > Usage

public:

void progressbar(float usage); //입력값을 시각적으로 바꾸어주는 함수

void setUsage(float input); //usage값을 set해줌

//progressbar을 구성하는 요소들

const char bar = '=';

const char blank = ' ';

const int LEN = 30;

float tick = (float)100 / LEN;

int barcount = Usage / tick;

Function 클래스는 콘솔창에 텍스트를 띄우는 것을 보조해주는 함수들을 가지고 있습니다.

구성 함수

private:

//float allcore;

int pos[12] = { 0,5,10,20,30,40,50,57,70,80,90,100 };//출력할 x좌표

public:

void to\_int(float usage);//usage를 소수점 없는 정수로 표현

void to\_float(float usage);//usage를 소수점 2자리까지 표현

void position(int x, int y);//x,y값을 입력받아 커서의 위치를 바꾸어줌

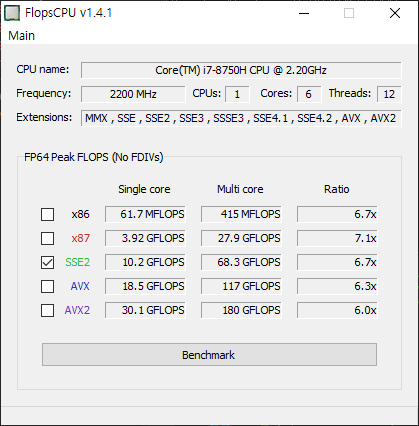
char\* Converter(CString unicode);//unicode를 multibyte로 변환

void ifusage(float usage);//usage의 값에 따라 색상 변경

void textcolor(int foreground, int background);//text의 색상을 변경

void progressbar(float usage);//Base의 progressbar 오버로드

void otherCPUscore();//다른 CPU의 점수출력

CPU 클래스는 사용자의 CPU의 정보를 읽어오고, CPU의 성능을 대략적으로 판별할 수 있는 Single Core와 Multi Core 벤치마크를 실행합니다.

왼쪽의 사진과 같이 x86의 명령어를 이용한 Single core, SSE2 명령어를 이용한 Multi core 점수가 제가 만든 프로그램에서 구한 값이 비슷하다는 것을 볼 수 있습니다.

x86: 61.7 / 58.71

SSE2: 68.3 / 68.45

콘솔창에서 왼클릭-정지/오른쪽클릭-재개

구성 함수

//float usage() { readcpu(); return \_fCPUv; }//소수점

void progressbar(float usage);//Base의 progressbar 오버로드

void setallcore(float all);//allcore을 set

void getallcore();//올코어 사용율 출력

void getCPUname();//CPU의 브랜드정보

//protected:

void readcpu();//CPU에서 정보를 읽어옴

float returnallcore() { readcpu(); return \_fCPUv; }//올코어 사용률을 리턴함

//float returncoregraph() { return allcore; }

void x86flops();//Single core flops 계산

void SSE2flops();//multi core flops 계산

double getAverage(const vector<double>& input);//

RAM 클래스는 시스템의 전체 RAM 사용량을 나타내줍니다.

구성 함수

void progressbar(float usage);// Base의 progressbar 오버로드

void RAMusage(); // 시스템의 램 사용량을 나타내줌

Engine 클래스는 모든 함수를 사용하여 프로그램을 만듭니다. main함수에서 이것을 호출합니다.

구성 함수

void EachCPU();//구현

~~Graph: 그래프를 그려주는 클래스~~ **구현 실패. 그래프가 움직이지 않아서 구현을 포기했습니다.**

~~구성 함수~~

~~void Graping();~~

~~void DrawGraph();~~

~~void line();~~

~~void lineUp();~~

~~void lineDown();~~

~~void line2();~~

~~void caldigit(float usage,int uptime);~~

**=프로젝트 후기=**

원래는 ALL CORE의 CPU 사용량을 하려고 했었습니다. 그렇게 했더니 너무 콘솔창이 횅해서 RAM 사용량을 먼저 추가하고, 다음 개별 코어의 사용량을 추가했습니다. 개별코어 사용량을 나타내는 과정에서, 마지막 코어는 올코어 사용량인데 갱신이 느리고 정수로 표현됩니다. 그래서 그 부분은 처음에 사용한 함수로 소수점 2번째까지 표현하도록 만들었습니다. 프로젝트를 만드는데 가장 큰 문제는 바로 각 정보의 갱신이였습니다. 함수가 중간에 끝나지 않기 때문에 갱신을 해주려면 정보를 갱신해주는 함수를 따로 만들어서 스레드로 만들어야 했습니다. 그래프를 그리는 함수를 만드려고 했는데, 결국 실패했습니다. 그래프를 왼쪽으로 계속 이동시켜주어야 하는데 그 기능을 만들지 못했습니다.

평소에 관심이 있던 HW분야를 주제로 잡고 프로젝트를 진행하여 좋았습니다. 다만 시간이 더 많고, 테스트할 수 있는 PC가 많았다면 더 정확한 flops를 계산할 수 있을 것 같아 아쉬웠습니다.