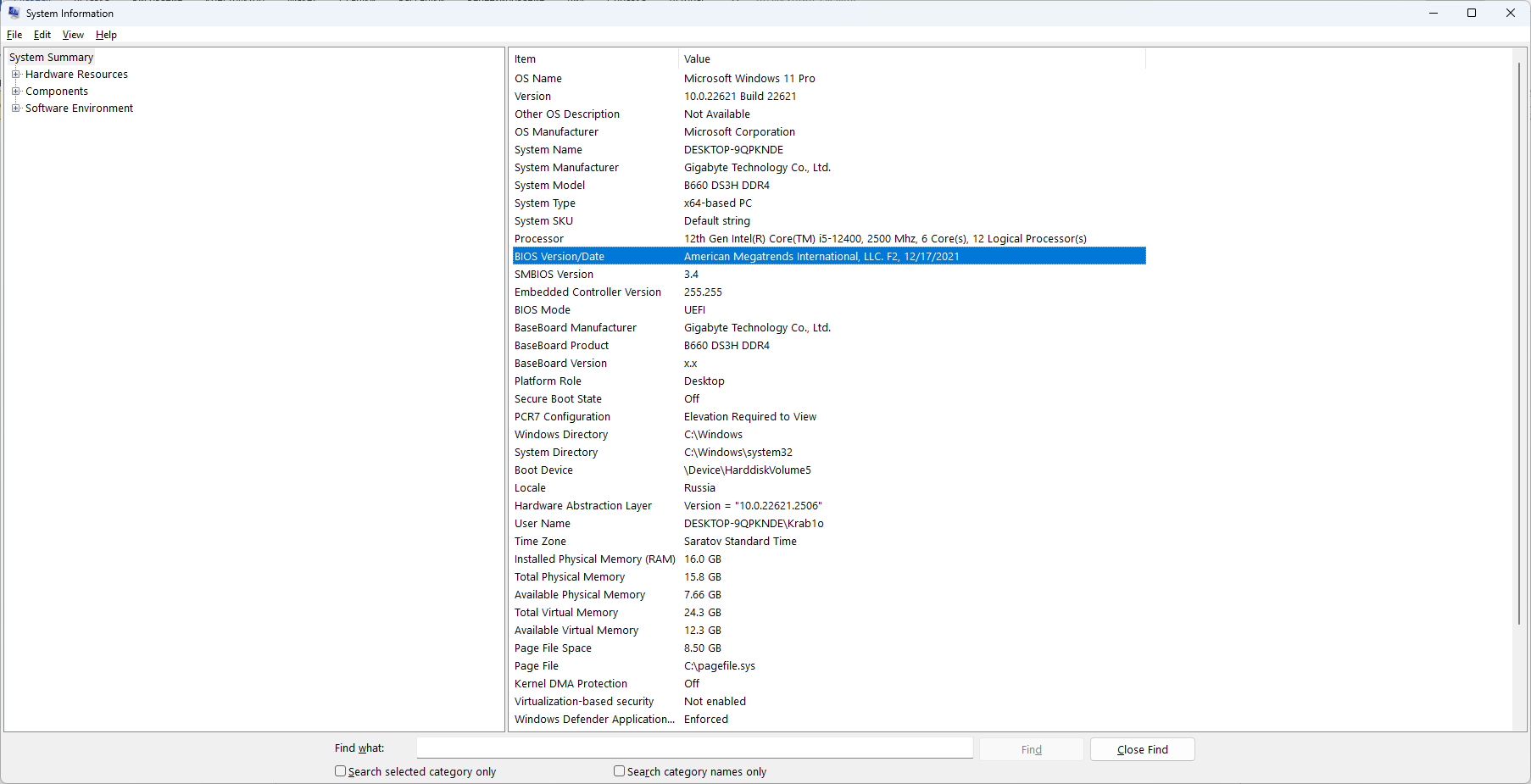
Задание выполнялось на компьютере со следующими параметрами:



Из важного:

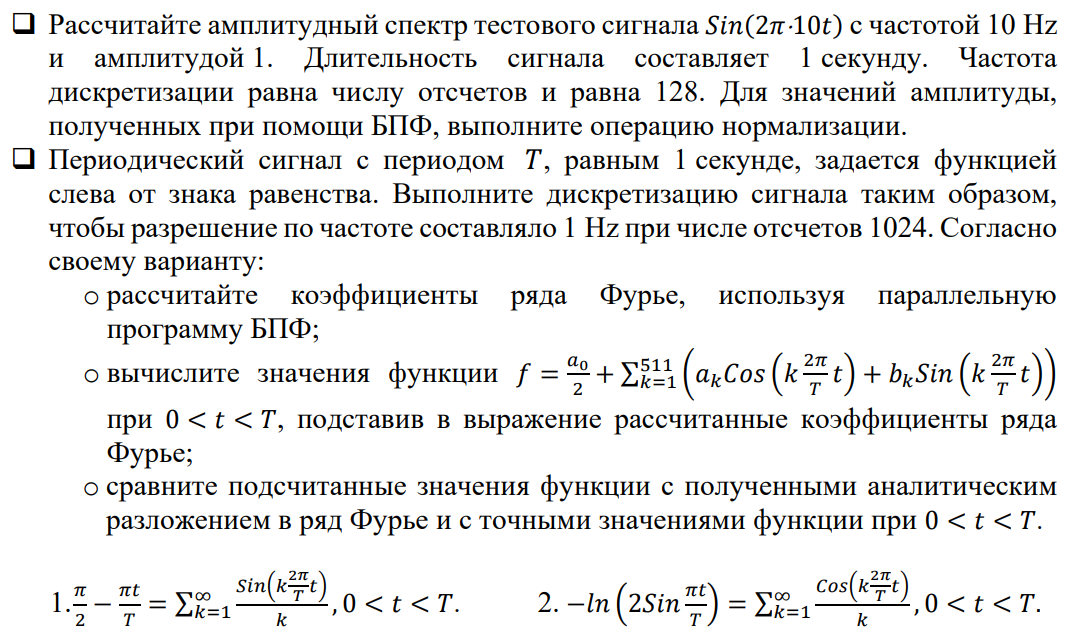
CPU: 12th Gen Intel(R) Core(TM) i5-12400, 2500 Mhz, 6 Core(s), 12 Logical Processor(s)

Motherboard: B660 DS3H DDR4

RAM: 16.0 GB со скоростью 2400MHz

**Задание 9, вариант 2**

**Предподготовка.**

****

**Код предподговки:**

void Pretreatment() {

int count = 128;

double count\_d = 128.0;

double eps = 1e-6;

complex<double>\* inputSignal = new complex<double>[count];

complex<double>\* outputSignal = new complex<double>[count];

double t;

for (int i = 0; i < count; i++) {

t = i / count\_d;

inputSignal[i] = complex<double>(sin(2.0 \* PI \* 10 \* t), 0);

}

int ampl;

ParallelFFT(inputSignal, outputSignal, count);

cout << "Pretreatment result:" << endl;

for (int i = 0; i < count; i++) {

if (abs(outputSignal[i].real()) > eps || abs(outputSignal[i].imag()) > eps) {

ampl = i;

cout << i << " " << outputSignal[i] << endl;

}

}

double norm = sqrt(outputSignal[10].imag() \* outputSignal[10].imag() +

outputSignal[10].real() \* outputSignal[10].real());

norm /= count / 2;

if (abs(norm - 1) < eps) {

cout << "OK" << endl;

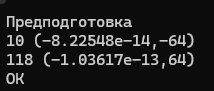
}

else {

cout << "ERROR" << endl;

}

}



**Вариант 2:**

double f(complex<double>\* signal, int size, double t) {

//вычисляем значение функции

double res = signal[0].real() / 2.0;

for (int i = 1; i < size / 2; i++) {

res += signal[i].real() \* cos(i \* 2.0 \* PI \* t / T)

- signal[i].imag() \* sin(i \* 2.0 \* PI \* t / T);

}

return res;

}

double StandardSum(double t) {

double eps = 1e-9;

double res = 0;

double s = 0;

int k = 1;

do {

s = sin(k \* 2.0 \* PI \* t / T) / k;

res += s;

k++;

} while (fabs(s) > eps);

return res;

}

void var\_1() {

int count = 1024;

complex<double>\* inputSignal = new complex<double>[count];

complex<double>\* outputSignal = new complex<double>[count];

for (int i = 1; i < count; i++) {

double t = i \* 1.0 / count;

inputSignal[i] = complex<double>((PI / 2 - PI \* t / T), 0);

}

//SerialFFT(inputSignal, outputSignal, count);

ParallelFFT(inputSignal, outputSignal, count);

for (int i = 0; i < count; i++) {

outputSignal[i] = outputSignal[i] / ((double)count / 2.);

}

cout << left << setw(10) << "Функция" << " "<< setw(10) << "Фурье" << " " << setw(10) << "Точное знач" << endl;

for (int i = 1; i < 64; i++) {

double t = i \* T / count;

cout << setw(10) << f(outputSignal, count, t) << " "

<< setw(10) << StandardSum(t) << " " << setw(10) << (PI / 2 - PI \* t / T) << endl;

}

}

int main()

{

setlocale(LC\_ALL, "rus");

Pretreatment();

var\_1();

}

