

 README.md

Кирюхин Андрей, БПИ191

Третье домашнее задание по Архитектуре вычислительных систем. Вариант 18. Исходники находятся в папке code.

Выполнение задания

Условие:

Задача о наследстве. У старого дона Энрике было два сына, у каждого из сыновей – еще по два сына, каждый из которых имел еще по два сына. Умирая, дон Энрике завещал все свое богатство правнукам в разных долях. Адвокат дон Хосе выполнил задачу дележа наследства в меру своих способностей. Правнуки заподозрили адвоката в укрывательстве части наследства. Требуется создать многопоточное приложение, которое при известных сумме завещания дона Энрике и доле каждого наследника, проверяет честность адвоката. При решении использовать принцип дихотомии.

Разработка:

Для решения была выбрана следующая модель построения многопоточных приложений: **Рекурсивный параллелизм** (используется в программах с одной или несколькими рекурсивными процедурами, вызов которых независим. Это технология «разделяй-и-властвуй» или «перебор-с-возвратами».)

После консультации с преподавателем, было решено генерировать доли наследников и размер наследства случайно внутри программы (в условии не было четких требований касательно этого).

- Основная рекурсивная функция

```
//Рекурсивная функция для формирования древа семьи для подсчета суммы полученной наследниками от Дона
void funcTree(int depth, double& result, int leftInt, int rightInt) {
    //depth = 3, мы дошли до 8 правнуков, останавливаем вызов новых функций дальше
    if (depth == 3) {
        //Параллельный подсчет суммы
        result = percentages[leftInt] * inheritance; //На данной глубине концы интервалов равны, поэтому берем одно из крайних значений
    }
    else {
        double l = 0, r = 0;

        //Передача в функцию интервалов, каждый раз разделяя их пополам
        int intLen = (rightInt - leftInt) / 2;
        thread left(funcTree, depth + 1, ref(l), leftInt, leftInt + intLen);
        thread right(funcTree, depth + 1, ref(r), rightInt - intLen, rightInt);

        //Ожидание завершения работы потоков
        left.join();
        right.join();

        result = l + r;
    }
}
```

В main в отдельном потоке запускается данная функция с параметрами *leftInt* = 0, *rightInt* = 7 (крайние значения индексов в массиве долей наследников).

Затем интервал делится на два и от них уже вызываются два раза данная рекурсивная функция. Такое деление происходит пока глубина рекурсии не равна трём, затем для каждого наследника считается полученная сумма и передается в родительскую функцию (так происходит для всех узлов). В итоге в первой вызванной рекурсивной функции (с параметрами 0,7) будет значение общей суммы, которую получили наследники.

- Функция генерации долей наследников

```

/*
    Генерация процентных долей наследников на основе "завещания" Дона.
    В этот момент адвокат может обмануть семью.
*/
double* generatePercentage() {
    bool isLawyerLies = rand() % 2 == 0; //С вероятностью 50 процентов адвокат обманет семью
    double* result = new double[8];

    double sum = 1;
    double tmp = (rand() % 20) / 100.0;
    result[0] = tmp;
    sum -= tmp;
    for (int i = 1; i < 7; i++)
    {
        tmp = (rand() % (int)(sum * 60)) / 100.0;
        if (isLawyerLies && tmp >= 0.2) {
            tmp -= 0.05; //Если адвокат решил обмануть наследников, он ворует 5 процентов у наследников, у которых доля >= 20%;
        }
        sum -= tmp;
        result[i] = tmp;
    }
    if (isLawyerLies) result[7] = sum / 2;
    else result[7] = sum;

    return result;
}

```

- Точка входа в программу

```

int main()
{
    srand(static_cast<unsigned int>(time(0)));
    setlocale(LC_ALL, "Russian");

    inheritance = rand() % 500000 + 500000;

    cout << "Дон Энрике оставил в наследство " << inheritance << "$" << endl;

    percentages = generatePercentage();
    cout << "Адвокат, на основе завещания Дона, посчитал следующее процентное распределение наследства среди наследников:" << endl;
    for (int i = 0; i < 7; i++)
    {
        cout << percentages[i] * 100 << "%, ";
    }
    cout << percentages[7] * 100 << "%" << endl << endl;

    //Подсчет суммы, которую наследники получили на руки
    double resultSum;
    thread main(funcTree, 0, ref(resultSum), 0, 7);
    main.join();

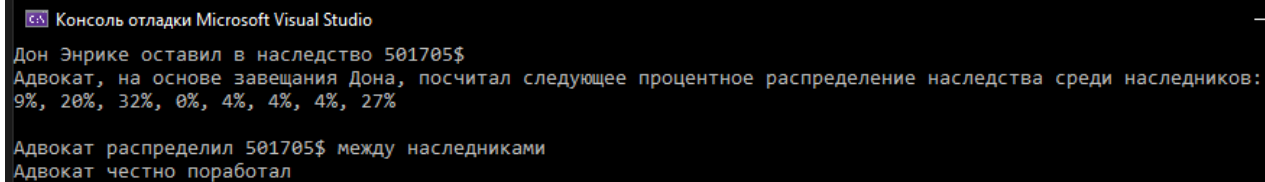
    //Выяснение обманул ли адвокат семью
    cout << "Адвокат распределил " << resultSum << "$ между наследниками" << endl;
    if (inheritance - resultSum < 0.001) {
        cout << "Адвокат честно поработал" << endl;
    }
    else {
        cout << "Адвокат обманул семью на " << inheritance - resultSum << "$!" << endl;
    }
}

```

Тестирование программы

В данном блоке проведен ряд тестов программы. И так как все параметры (размер наследства, доли) генерируются случайно внутри программы, то продемонстрирую работоспособность программы при разных сценариях.

- Адвокат не обманул семью

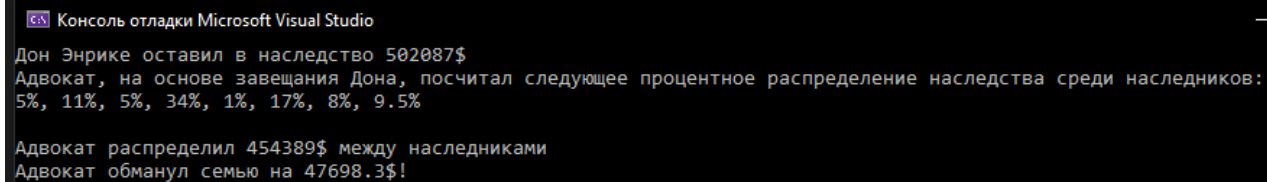


Консоль отладки Microsoft Visual Studio

Дон Энрике оставил в наследство 501705\$
Адвокат, на основе завещания Дона, посчитал следующее процентное распределение наследства среди наследников:
9%, 20%, 32%, 0%, 4%, 4%, 4%, 27%

Адвокат распределил 501705\$ между наследниками
Адвокат честно поработал

- Адвокат обманул семью



Консоль отладки Microsoft Visual Studio

Дон Энрике оставил в наследство 502087\$
Адвокат, на основе завещания Дона, посчитал следующее процентное распределение наследства среди наследников:
5%, 11%, 5%, 34%, 1%, 17%, 8%, 9.5%

Адвокат распределил 454389\$ между наследниками
Адвокат обманул семью на 47698.3\$!

Таким образом была успешно разработана и протестирована данная программа.