

funbox

Квалификационные задания для C разработчиков

О заданиях

Предлагаемый набор задач позволяет нам разносторонне оценить ваши навыки.

Первое задание позволит убедиться, что вы не только хорошо владеете спецификациями C/C++, но и представляете, что лежит "под капотом" той или иной языковой конструкции. Вторая задача докажет, что вы владеете примитивами Unix-программирования, а успешное выполнение третьей убедит нас в том, что вы отлично разбираетесь в алгоритмах и структурах данных, а также неплохо оптимизируете свой код.

Если вы не можете предоставить законченный вариант, но у вас есть прототип, или по крайней мере внятная идея того, как должно выглядеть решение - напишите нам. Но разумеется, мы будем очень рады человеку, который предоставит нам хорошо закомментированный и оттестированный код для каждой из задач.

Ответы на задания вы можете составить в свободной форме и прислать нам вместе с вашим резюме (не забыв указать вакансию *C разработчика*) на почтовый ящик wanted@fun-box.ru. После проверки заданий, мы обязательно сообщим вам о нашем решении.

В общем, это всё. Перейдя на следующую страницу, вы увидите наши вопросы и задания.

Task I

Известно, что многие возможности высокоуровневых языков программирования могут быть с той или иной степенью полноты имитированы на "низкоуровневых" языках. Так, на языке C++, лишенном встроенных средств сборки мусора, можно использовать "умные указатели". То же справедливо для лямбда-функций и замыканий, которые имитируются как библиотекой boost, так и другими. Мы предлагаем вам решить задачу "портирования" одной из ключевых особенностей C++ - обработки исключений - на язык C. Решение сводится к созданию набора макросов, схожих (но не обязательно идентичных) соответствующим ключевым словам языка C++.

Пример использования наиболее примитивной реализации:

```
#include <stdio.h>
#include "pure-c-exceptions.h"

int div(int a, int b)
{
    return a/b;
}

int main()
{
    int c;
    __TRY__
    {
        c = div(2,0);
        printf("эта команда не будет выполнена :( \n");
    }
    __CATCH__
    {
        printf("деление на 0!\n");
    }
    printf("я все ещё жив!");
    return 0;
}
```

Ваши макросы могут содержать вызовы функций, в том числе системных, использовать глобальные переменные. Постарайтесь реализовать вариант, более близкий к возможностям C++, чем приведенный выше.

В чем отличие кроссплатформенного варианта от реализации, "заточенной" под Windows? Какие преимущества вы видите в каждом из них?

Task II

Реализуйте собственную интерактивную командную оболочку (*shell*), которая удовлетворяет следующим условиям:

- возможность редактирования командной строки так, как это позволяет делать *unix shell*
- возможность навигации по истории команд клавишами управления, как это реализовано в *unix shell*
- возможность выполнения внешних команд. Например, команда `myshell> ls ~` должна отобразить список файлов домашней директории пользователя
- возможность перенаправления потоков ввода-вывода:
`myshell> sort < input.txt > output.txt`
- возможность организовать конвейер команд:
`myshell> ls | grep '.sh' | sort > shlist`

Task III

У вас есть массивы точек A1 и A2. Для каждой точки указаны её координаты на поверхности Земли: широта (latitude) и долгота (longitude), заданные дробным числом. Например:

`A1[0].lat = 59.934228; A1[0].lon = 30.324594;` - координаты Казанского Собора в Санкт-Петербурге

`A1[1].lat = 40.689167; A1[1].lon = -74.044583;` - статуя Свободы в Нью-Йорке

Реализуйте алгоритм, который для каждой точки из массива A2 найдет расстояние до ближайшей к ней из массива A1. Массивы A1 и A2 могут содержать большое количество элементов (десятки и сотни тысяч), поэтому требуется очень высокая скорость обработки. Хорошая точность также имеет значение.

Обоснуйте, почему реализованный вами способ является оптимальным по соотношению скорость/точность. Какие модификации вы бы произвели, чтобы пожертвовать скоростью ради более точного определения расстояний (погрешность - не более метра)? Какие модификации позволят нарастить скорость в ущерб точности (допуская погрешность в несколько километров)?

Спасибо за время потраченное на выполнение заданий!