

Лабораторная работа № 2 по курсу дискретного анализа: Словарь

Выполнил студент группы 08-208 МАИ *Левитанов Денис*.

Условие

1. Необходимо создать программную библиотеку, реализующую указанную структуру данных, на основе которой разработать программу-словарь. В словаре каждому ключу, представляющему из себя регистронезависимую последовательность букв английского алфавита длиной не более 256 символов, поставлен в соответствие некоторый номер, от 0 до 264 - 1. Разным словам может быть поставлен в соответствие один и тот же номер.
2. Б-дерево.

Метод решения

В цикле считывать слова, подающиеся на вход, проверить первый символ, если он равен «+», то считать пара «ключ значение» в класс TElem, и этот элемент вставить в Б-дерево. Если такой ключ уже существует вывести «Exist». При «-» считать ключ и удалить его в дереве, если такого ключа в дереве нет, вывести «NoSuchWord». При «!» считать следующее слово, если это «Save», то считать имя файла и сериализовать существующее дерево. Если «Load» то считать имя файла, создать новое дерево и десериализовать данные из файла. При неудаче работа продолжается с исходным деревом. Если первый символ введенного слова не совпадает ни с одним из вышеперечисленных, то выполнить поиск этого слова в дереве. Также на протяжении всей программы обрабатываются и выводятся такие ошибки как: memory error, file error, deserialization error.

Описание программы

1. *Заголовочные файлы:*
TVector.h(описание и реализация собственного вектора)
TSmartPointer.h(описание и реализация собственных умных указателей)
TNode.h(Объявление узла Б-дерева)
TBTree.h(Объявление Б-дерева).
2. *Cpp файлы:*
main.cpp(Считывание и вывод данных, управление программой)
TBTree.cpp (Реализация вставки, удаления, поиска элемента, сериализации, десериализации дерева)
TNode.cpp(Реализация конструкторов узла).

Дневник отладки

Программа не проходила тест №4 из-за ошибки в алгоритме удаления. На тесте №8 программа выдавала неверный ответ из-за считывания данных не в адрес строки, а в адрес указателя. На 11 теста был получен runtime error signal 6 по нехватке памяти, из-за считывания строки переменной длины в переменную с выделенной памятью в 257 символов. Проблема была решена динамическим выделением памяти под строку. На 14 тесте программа не проходила по времени, ускорить программу получилось переписав поиск элемента в узле с обычного линейного на бинарный.

Тест производительности

1. 200 000 строк без сериализации:
Time of working: 1.393
2. 400 000 строк без сериализации:
Time of working: 2.787
3. 50 000 строк с сериализацией:
Time of working: 6.279
4. 100 000 строк с сериализацией:
Time of working: 25.568

Выводы

Б-дерево - это сбалансированное, сильно ветвистое дерево. Дерево я реализовывал таким образом, что оно целиком хранится в оперативной памяти, но обычно Б-дерево используется во внешней памяти, то есть каждый узел записывается на диск, а по мере необходимости нужные узлы считываются с диска, и в итоге в оперативной памяти находится только рабочий узел, что помогает работать с большими объемами данных, не забивая оперативную память.