

Факультет программной инженерии и компьютерной техники

Низкоуровневое программирование

Лабораторные работа №1

Вариант 4

Преподаватель: Кореньков Юрий

Выполнил: Жданок Прокопий P33102

# Выполнение

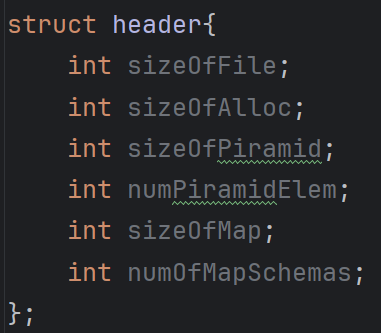
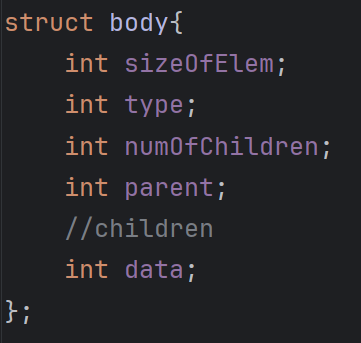
Цель: разработать хранилище данных большого объема, в котором хранится древовидная информация. При работе с файлом нужно использовать О(1) памяти, О(1) вставка, О(n) удаление, обновление.

Описание работы: В этой работе я использовал 3 основных файла: fileStruct, reader, alloc. Reader управляет ммапингом.

Alloc помнит о свободных местах в файле, чтобы быстро вставлять данные.

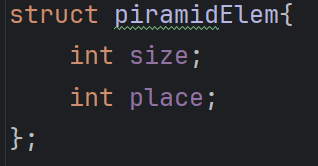
fileStruct знает о строении файла и основных интерфейсах их обработки.

Аспекты реализации: Файл состоит из header, в котором есть вся необходимая информация для обхода файла. После хедера идут struct body, которые ссылаются друг на друга через адрес начала файла или бакета.

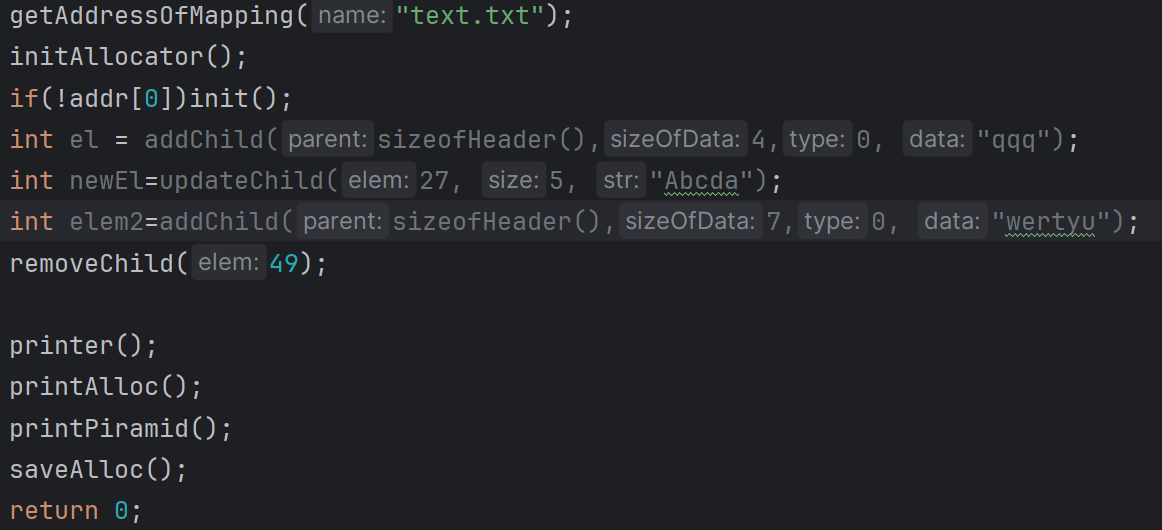
 

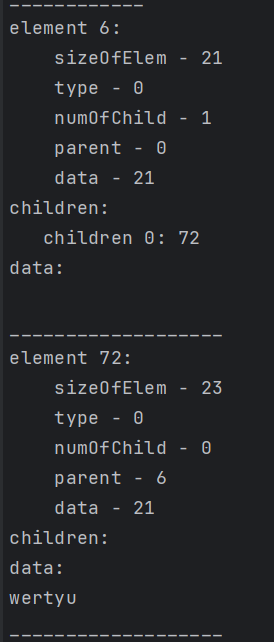
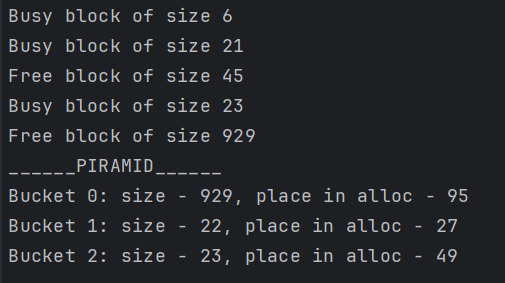
Чтобы не использовать слишком много оперативной памяти, я ммаплю только несколько киллобайт, если мне нужны данные из другого блока, тогда я заново выполняю ммап, а старую часть затираю. В один момент мне может понадобиться до трех бакетов в файле(родитель, элемент, ребенок). Родитель всегда один, обрабатываемый элемент в данной операции всегда один, а детей можно обрабатывать в цикле, в начале которого проверять бакет. Таким образом reader имеет 3 адреса, которые по необходимости меняет.

Вернемся к аллокатору. В нем лежит пирамида, которая хранит пустые куски(пирамида всегда держит в своей вершине самый большой кусок). Инзачально там, условно, 1024 байта свобюодной памяти. Когда приходит запрос на, например, 100 байт, аллокатор отщипывает от самого большого куска, а остальное кладет обратно в пирамиду. Если запросили больше места, чем есть, то на верх пирамиды кидается элемент, равный удвоенному предыдущему размеру. Таким образом мы имеет довольно классную ассимптотику, которая зависит от количества удалений и файла, однако уменьшается, когда мы добавляем объекты.



# Тесты



Результаты: Всегда, когда я использую mmap или malloc, я в этой же функции освобождаю место назад, поэтому у меня нет утечек памяти. Вставка происходит за константу, так как мы просто берем элемент с вершины аллокатора и размещаем туда наш объект.

# Вывод

Выводы: в данной работе я должен был задуматься о внутреннем строении больших хранилищ информации, их аллокации и чтении/записи.