

Работа 4. Изучение особенностей создания динамических списков и структур

Цель работы

Изучить особенности использования структур в языке C, а также научиться применять их для создания динамических списков; познакомиться с примером служебной структуры, используемой POSIX-совместимыми операционными системами для агрегирования информации об элементе каталога файловой системы.

Теоретический материал

Для выполнения работы вам будет необходимо использовать материал по следующим темам:

1. Структуры и указатели на структуры

Структуры в C послужили прообразом классов (как концептуально, так и в плане синтаксиса) во многих объектно-ориентированных языках, таких как C++, C# и Java – но в отличие от классов, структура содержит в себе только информационные поля (переменные-члены структуры), не содержит методы (функции), и не предусматривает ограничения доступа.

Воспользуйтесь следующими источниками информации, чтобы разобраться с использованием структур в C:

- Общая информация о работе со структурами: <https://metanit.com/c/tutorial/6.1.php>
- Указатели на структуры: <https://metanit.com/c/tutorial/6.3.php>
- Особенности размещения структур и их полей в памяти (дополнительный материал): <https://metanit.com/c/tutorial/6.9.php>

2. Работа с каталогами файловой системы (директориями)

Структуры активно используются для хранения агрегированных данных в операционных системах (этому не приходится удивляться, учитывая что все мэйнстримные операционные системы написаны на C). В этой работе мы познакомимся с примером такой структуры - структурой `dirent` (от английского `directory entry` - элемент директории).

Как следует из названия этой структуры, она содержит информацию об элементе директории (каталога) в файловой системе. Как вы знаете из опыта навигации по файловой системе, этот элемент может быть файлом либо поддиректорией.

Чтобы выполнить часть задания, связанную с файловой системой, вам понадобится ознакомиться со следующими источниками:

- Общая информация о работе со структурами `dirent` для доступа к оглавлению файловой системы на наглядном примере: <https://firststeps.ru/linux/r.php?20>
- Более подробная справка по структурам `dirent` и связанным с ними функциям: https://en.wikibooks.org/wiki/C_Programming/POSIX_Reference/dirent.h

3. Работа с динамическими списками

Одним из самых частых применением структур является использование их для создания динамических списков. Это список, элементы которого являются выделенными в динамической памяти структурными переменными. Структура, переменными-экземплярами которой являются элементы списка, содержит какие-то поля данных (или указатель на объект данных), а также один или два указателя на соседние элементы списка. В результате, элементы списка не обязательно находятся в памяти друг за другом, но благодаря тому что они объединены в цепочку указателей, с ними можно работать как с последовательной коллекцией элементов (выполнять поиск нужного элемента, менять элементы местами, просто изменив указатели, и др.). Если в каждом элементе есть только один указатель на соседний элемент списка (следующий элемент), такой список называется *односвязным*. Если в каждом элементе содержатся два таких указателя (на следующий и на предыдущий элемент списка) – это *двусвязный* список. Двусвязные списки удобнее при поиске и сортировке, поскольку от текущего элемента можно “двигаться” по списку в обе стороны, проходя по указателям. Односвязные списки можно по понятной причине проходить только от начала к концу – но зато они проще в реализации, а манипуляции со списком (добавление, удаление, перестановка элементов) требуют меньше изменений в указателях. Последнее важно, поскольку типичная ошибка при работе со списком – нарушение очередности переприсваивания указателей, в результате чего часть списка теряется.

Для ознакомления с реализацией динамических списков в С воспользуйтесь следующим материалом:

- Реализация односвязного списка на С: https://learn.c.info/adt/linked_list.html
- Реализация двусвязного списка на С: https://learn.c.info/adt/double_linked_list.html

Задание

1. Реализовать простой связный список на языке С с поддержкой следующих операций:
 - a. Вставка нового элемента
 - b. Поиск элемента по названию
 - c. Удаление элемента
2. В качестве элемента данных в динамическом списке использовать структурную переменную, в полях которой должна храниться информация о файлах и папках, содержащихся в жестко заданном программе конкретном каталоге файловой

системы. В структурной переменной должны храниться имя объекта (файла или подкаталога), а также два информационных атрибута этого объекта, в соответствии с индивидуальным заданием – например, дата создания файла/подкаталога и его inode (в Unix-подобных системах).

3. Запрашивать у пользователя в цикле одну из трёх операций:
 - a. просканировать информацию о содержимом каталога и внести ее в список (или обновить, если список уже существует – при этом для простоты не обновлять информацию в элементах списка, считая ее константной, а только использовать удаление и вставку элементов, если в каталоге появились новые объекты или были удалены старые)
 - b. вывести список элементов на экран (выводить только имена файлов/подкаталогов без дополнительных атрибутов)
 - c. вывести информацию о конкретном объекте из элемента списка со всеми его атрибутами, сохраненными в элементе списка.
4. ***(задание повышенной сложности)** Дополнить программу “списком списков”, в котором должна храниться история сканирований содержимого каталога (мгновенные снимки списка элементов каталога); дополнять “список списков” новым списком в тот момент времени, когда пользователь дает команду просканировать/обновить информацию о содержимом каталога. При выводе списка файлов на экран дополнительно запрашивать, из какого мгновенного снимка пользователь хочет получить информацию.