

Семинар 11. Файловые системы. Часть 2.

Специальные функции для работы с содержимым директорий. Стандартные системные вызовы `open()`, `read()` и `close()` не могут помочь программисту изучить содержимое файла типа "директория". Для анализа содержимого директорий используется набор функций из стандартной библиотеки языка C.

С точки зрения программиста в этом интерфейсе директория представляется как файл последовательного доступа, над которым можно совершать операции чтения очередной записи и позиционирования на начале файла. Перед выполнением этих операций директорию необходимо открыть, а после окончания - закрыть.

Для открытия директории служит функция `opendir()`, которая подготавливает почву для совершения операций и позиционирует нас на начале файла. Чтение очередной записи из директории осуществляет функция `readdir()`, одновременно позиционируя нас на начале следующей записи (если она, конечно, существует). Для операции нового позиционирования на начале директории (если вдруг понадобится) применяется функция `rewinddir()`. После окончания работы с директорией ее необходимо закрыть с помощью функции `closedir()`.

Понятие о файлах, отображаемых в память (memory mapped файлах).
Системные вызовы `mmap()`, `munmap()`. Мы уже говорили, что с помощью системного вызова `open()` операционная система отображает файл из пространства имен в дисковое пространство файловой системы, подготавливая почву для осуществления других операций. С появлением концепции виртуальной памяти, которую мы обсуждали на соответствующей лекции, когда физические размеры памяти перестали играть роль сдерживающего фактора в развитии вычислительных систем, стало возможным отображать файлы непосредственно в адресное пространство процессов. Иными словами, появилась возможность работать с файлами как с обычной памятью, заменив выполнение базовых операций над ними с помощью системных вызовов на использование операций обычных языков программирования. Файлы, чье содержимое отображается непосредственно в адресное пространство процессов, получили название файлов, отображаемых в память, или, по-английски, memory mapped файлов. Надо отметить, что такое отображение может быть осуществлено не только для всего файла в целом, но и для его части.

С точки зрения программиста работа с такими файлами выглядит следующим образом:

1. Отображение файла из пространства имен в адресное пространство процесса происходит в два этапа: сначала выполняется отображение в дисковое пространство, а

уже затем из дискового пространства в адресное. Поэтому вначале файл необходимо открыть, используя обычный системный вызов `open()`.

2. Вторым этапом является отображение файла целиком или частично из дискового пространства в адресное пространство процесса. Для этого используется системный вызов `mmap()`. Файл после этого можно и закрыть, выполнив системный вызов `close()`, так как необходимую информацию о расположении файла на диске мы уже сохранили в других структурах данных при вызове `mmap()`.

3. После этого с содержимым файла можно работать, как с содержимым обычной области памяти.

4. По окончании работы с содержимым файла, необходимо освободить дополнительно выделенную процессу область памяти, предварительно синхронизировав содержимое файла на диске с содержимым этой области (если , конечно, необходимо). Эти действия выполняет системный вызов `munmap()`.