Оглавление

ЛАБОРАТОРНАЯ РАБОТА 7 ФАЙЛОВАЯ СИСТЕМА

Теория

Open

Перед использованием файла процесс должен сначала открыть его для чтения или записи. Системный вызов ***open*** имеет следующий синтаксис:

```
#include <fcntl.h>

int fd=open(const char *path, int oflag, mode_t mode);
```

где path- это абсолютное или относительное имя файла, а mode указывает на права, ассоциируемые с файлом при его создании.

Флаги, передаваемые при помощи oflag, указывают на необходимость открытия файла после его создания на:

- чтение (read) – O_RDONLY;
- запись (write) – O_WONLY;
- чтение/запись (read/write) – O_RDWR;
- добавление (append) – O_APPEND;
- и т. д.

В случае неудачи ***open*** возвратит -1 (errno будет содержать код ошибки), а в случае удачи будет возвращен дескриптор файла.

Каждый процесс имеет *маску создания файлов*, определенную по умолчанию, являющуюся битовой маской прав, не предоставляемых создаваемым файлам. Когда пользователь указывает параметр mode для функции ***open*** или ***creat***, пользователь обнуляет разряды, указанные в маске по умолчанию. Для изменения маски, принятой по умолчанию, используется системный вызов ***chmod***.

Creat

Для создания файла также применяется системный вызов ***creat***, который эквивалентен вызову ***open*** с флагами O_WRONLY, O_CREAT и O_TRUNC.

#include <fcntl.h>

int ***creat***(const char *path, mode_t mode);

как и в случае ***open*** path- это абсолютное или относительное имя файла, а mode устанавливаемые права доступа к файлу.

Close

Функция ***close*** разрывает связь между файловым дескриптором и открытым файлом, созданную функциями ***creat, open, dup*** и ***fcntl.***

#include <fcntl.h>

int ***close***(int fildes);

В случае неудачи ***close*** возвратит -1 (errno будет содержать код ошибки), а в случае удачи будет возвращен 0.
Функция ***exit*** автоматически закрывает открытые файлы.

Dup

Функция ***dup*** используется для дублирования существующего файлового дескриптора:

int ***dup(***int fildes);

Файловый дескриптор должен быть предварительно получен с помощью ***creat, open*** и ***dup.*** В случае успешного завершения функция возвращает новый файловый дескриптор, свойства которого идентичны свойствам дескриптора fildes.

Lseek

С файловым дескриптором связан файловый указатель, определяющий текущее смещение в файле, начиная с которого будет произведена последующая операция чтения или записи. С

помощью ***lseek*** можно установит файловый указатель на любое место файла;

#include <unistd.h>

off_t ***lseek***(int fildes, off_t offset, int whence);

Интерпретация аргумента offset зависит от аргумента whence, который может принимать следующие значения:

- SEEK_CUR. Указатель смещается на offset от текущего положения;
- SEEK_END. Указатель смещается на offset байт от конца файла;
- SEEK_SET. Указатель устанавливается равным offset.

Read и Write

Вызовы ***read*** и ***write*** имеют одинаковую семантику, поэтому приведем пример только одной из этих функций:

#include <unistd.h>

ssize_t nread=***read***(int fd, void *buf, size_t count);

где fd- это дескриптор файла, buf- указатель на буфер в пользовательском адресном пространстве, в который должно производиться чтение данных, а параметр count- количество считываемых байтов.

Возможно возникновение ситуации, когда количество считываемых байтов меньше, чем величина count. Это может произойти при

достижении конца файла или в случае отсутствия доступных данных при обращении к файлам FIFO или устройствам.

Ответственность за емкость буфера buf, достаточную для помещения count байтов данных, несет пользователь. Вызов ***read*** возвращает количество переданных байтов (nread). Эта функция также производит смещение указателя на nread байтов, поэтому следующий вызов ***read*** или ***write*** начнет передачу данных с точки завершения действий предыдущей функции.

Файл может быть открыт в режиме добавления, для чего системному вызову open необходимо передать флаг O_APPEND. В этом случае перед вызовом write для указанного дескриптора ядро установит указатель смещения на конец файла. Если пользователь открывает файл в режиме добавления, то это не повлияет на операции с этим файлом, производимые через другие дескрипторы.

Fcntl

После открытия файла и получения файлового дескриптора процесс может производить различные файловые операции. Функция ***fcntl*** позволяет процессу выполнить ряд действий с файлом, используя его дескриптор:

#include <fcntl.h>

int ***fcntl(***int fildes, int cmd,…);

Функция выполняет действие cmd с файлом, а третий аргумент зависит от конкретного действия:

- F_GETLK. Проверить существование блокирования записи файла. Блокирование описывается структурой flock, указатель на которую передается в качестве третьего аргумента;
- F_SETLK. Установить блокирования записи файла. Структура flock описывает блокирование и указатель на нее передается в качестве третьего аргумента;
- F_SETLKW. Аналогично предыдущему, но при невозможности блокирования по причине уже существующих блокировок, процесс переходит в состояние сна, ожидая пока последние освободятся.

Чему нужно научиться

Изучить основные системные вызовы для работы с файлами.

Задание

Уровень 1, 2, 3 (А)

Необходимо чтобы два процесса заносили очередной элемент в связанный список, который хранится в файле.

ПРИЛОЖЕНИЕ

Подключение файловых систем (монтирование)

Пользователи видят дерево каталогов и файлов как единую структуру. На самом деле различные каталоги этого дерева могут соответствовать различным разделам диска, находиться на различных дисках и даже на других компьютерах. Раздел диска присоединяется к дереву в каталоге, называемом *точкой подключения* или *монтирования* (*mount point*). *Точка подключения* и все расположенные ниже ее каталоги образуют файловую систему.

Для монтирования файловой системы в дерево каталогов LINUX&UNIX необходимо иметь раздел на диске, CD-ROM или гибкий диск, который вы хотите подключить. Следует убедиться также, что каталог (*точка подключения*), к которому вы хотите подключить файловую систему, действительно существует. Подключение файловой системы этот каталог не создает, он должен существовать до попытки подключения, иначе подключение окончится неудачей.

Для подключения файловых систем вручную используют команду:

mount [-t type] [-o options] device mountpoint

где device – физическое устройство, которое необходимо подключить;

а mountpoint – точка подключения.

Использовать команду **mount** может только супер пользователь (root).

-f	Имитирует подключение файловой системы. Выполняет все действия, кроме системного вызова для настоящего подключения
-v	Подробный отчет. **mount** предоставляет дополнительную информацию о своих действиях
-wr	Подключает файловую систему с доступом для чтения и записи
-ro	Подключает файловую систему с доступом только для чтения
-t type	Указывает тип подключаемой файловой системы.

	Допустимыми являются типы: ext2,msdos,hpfs,nfs,iso9660
-a	Указывает **mount** подключить все файловые системы, перечисленные в файле /etc/fstab
-o options list	Указывает mount применить список опций к подключаемой файловой системе

Таблица 1 Опции mount

Например, следующая команда подключает CD-ROM SCSI только для чтения с форматом файлов ISO 9660 в каталог /mnt/cdrom:

mount –r –t iso9660 /dev/sr0 /mnt/cdrom

А следующая команда подключает все файловые системы Network File System(nfs), перечисленные в файле /etc/fstab:

mount –vat nfs

Рассмотрим монтирование гибкого диска. Сначала выполните команду **ls** /mnt/floppy и убедитесь в том, что этот каталог пуст.
Затем подключите гибкий диск в эту точку, используя команду **mount**. Учитывайте, что если файловая система, которую пользователь хочет монтировать, определена в файле /etc/fstab, можно задать только точку монтирования, остальное будет извлечено командой **mount** из файла /etc/fstab. Снова выполните команду **ls** /mnt/floppy и убедитесь, что содержимое дискеты теперь появилось в этом каталоге.
Если правильно подключить файловую систему не удается, воспользуйтесь командой **mount** –vf device mountpoint. Эта команда

выполнит все действия, кроме подключения, и выдаст подробный отчет.

Для подключения файловых систем при загрузке используют конфигурационный файл /etc/fstab (file system table). Список используемых файловых систем изменяется редко, поэтому удобно подключать их при загрузке. Файловые системы перечисляются по одной в строке. Поля в строках разделяются пробелами или табуляцией.

Файловая система	Подключаемое блочное устройство или удаленная файловая система
Точка подключения	Куда подключить файловую систему. Чтобы сделать систему невидимой в дереве каталогов используйте слово none
Тип	Указывает тип подключаемой файловой системы (*)
Опции подключения	Разделенный запятыми список опций должен содержать, по крайней мере, тип подключения
Периодичность резервного копирования	Указывает, как часто следует выполнять резервное копирование с помощью команды dump. Если это поле отсутствует, то файловая система не нуждается в резервном копировании
Номер прохода	Задает порядок проверки целостности файловых систем при загрузке с помощью команды **fsck**. Для корневой файловой системы надо указывать 1, для остальных 2.

	Если значение не указано, то целостность файловой системы при загрузке проверяться не будет

Таблица 2 Поля файла fstab

(*) В настоящее время поддерживаются файловые системы следующих типов:

- Minix – локальная файловая система с именами файлов до 14 или 30 символов;
- Ext2 - локальная файловая система с длинными именами файлов и другими возможностями;
- Msdos - локальная файловая система для разделов MS DOS;
- Hpfs - локальная файловая система для разделов OS/2 (High Performance File System);
- Iso9660 - локальная файловая система, используемая с CD-ROM;
- Nfs - файловая система для подключения разделов удаленных систем;
- Swap – раздел или файл подкачки.

Приведем пример fstab:

# device	directory	type	options
/dev/hda1	/	ext2	defaults
/dev/hda2	/usr	ext2	defaults
/dev/hda2	none	swap	sw

Слово defaults указывает, что при подключении файловой системы следует применить набор опций по умолчанию:

- Файловую систему следует подключить с разрешенным доступом для чтения и записи;
- Она будет рассматриваться как отдельное блочное устройство;
- Весь файловый ввод/вывод будет выполняться асинхронно;
- Разрешено выполнение программных файлов;
- Файловая система может подключаться с помощью команды **mount** –a;
- Биты UID и GID интерпретируются в этой файловой системе;
- Обычным пользователям не разрешено подключать эту файловую систему.

Отключение файловых систем

Для отключения файловых систем используют команду:

umount device или mountpoint

где device – физическое устройство, которое необходимо отключить; а mountpoint – точка подключения.

Отключает все файловые системы команда **umount** –a.
Команда **umount** –t fstype отключает только файловые системы указанного типа.
Рассмотрим размонтирование гибкого диска:
cd /mnt/floppy
pwd
ls -l

Попробуйте отключить гибкий диск, используя команду **umount**. Учитывайте, что, так как файловая система, которую вы хотите отключить, используется в этот момент, то появится сообщение об ошибке. Перейдите в каталог другой файловой системы и попробуйте снова.

Сетевая файловая система

Сетевая файловая система (***Network File System, NFS***) позволяет подключать файловые системы других компьютеров, используя протокол TCP/IP. Под NFS файловая система удаленного компьютера подключается и выглядит для пользователей как локальная файловая система. Для этого необходимо:

- Компьютеры должны быть способны связаться друг с другом по протоколу TCP/IP;
- Компьютер с файловой системой, которую хотим подключать по сети, должен предоставить ее для подключения. Такой компьютер называется *сервером*, а процесс предоставления – *экспортом файловой системы;*
- Компьютер, подключающий экспортированную файловую систему, должен подключить ее как файловую систему типа NFS через файл /etc/fstab во время загрузки или вручную с помощью команды **mount**. Такой компьютер называется *клиентом.*

Перед тем как файловая система будет предоставлена для сетевого подключения, она должна быть смонтирована на сервере, т.е. подключена локально.

Экспорт файловой системы

На сервере должны быть запущены демоны NFS **rpc.mountd** и **rpc.nfsd**. Их следует запускать, после того как был запущен **portmap.**

Запись об экспортируемой файловой системе должна быть внесена в конфигурационный файл /etc/export. Этот файл используется демонами **rpc.mountd**, **rpc.nfsd**, чтобы определить какие файловые системы экспортируются и какие для них существуют ограничения. Файловые системы перечисляются в этом файле по одной в строке. В начале каждой строки указывается имя точки подключения локальной файловой системы, а затем список компьютеров, которым разрешается доступ к ней. После имени компьютера в круглых скобках через запятую может следовать список опций подключения.

insecure	Разрешается доступ с этого компьютера без аутентификации
secure	Для доступа с этого компьютера требуется аутентификация
Root_squash	Все запросы от root с UID=0 на клиенте отображаются на UID nobody на сервере
no_root_squash	Запросы от UID 0 не отображаются
ro	Файловая система подключается только для чтения
rw	Файловая система подключается для чтения и записи
all_squash	Все UID и GID отображаются на анонимного пользователя

Таблица 3 Опции подключения, допустимые в /etc/export

Приведем пример файла export:

```
/home          white.tristar.com(rw) red.tristar.com(rw)
/projects      brown.tristar.com(ro)
/public        (ro,insecure,root_squash)
```

Список компьютеров, которым разрешен доступ к /public не указан. Это означает, что подключать ее может любой.

Подключение файловой системы NFS

Подключение можно выполнить через файл /etc/fstab во время начальной загрузки или вручную с помощью команды **mount**. Имя файловой системы должно быть указано в следующем формате:

Servername:filesystemname

где Servername – имя сервера, экспортирующего файловую систему;
а Filesystemname – имя файловой системы на сервере.

rsize=n	Задает размер датаграмм (в байтах), используемых клиентом NFS в запросах на чтение. По умолчанию 1024байта
wsize=n	Задает размер датаграмм (в байтах), используемых клиентом NFS в запросах на запись. По умолчанию 1024байта
timeo=n	Устанавливает время ожидания выполнения запроса клиентом NFS. По умолчанию 0,7сек
hard	Выполняет жесткое подключение этой файловой системы
soft	Выполняет мягкое подключение этой файловой системы

intr	Разрешает прерывать вызов NFS

Таблица 4 **Опции подключения, применяемые NFS**

Пример fstab:

device directory type options
mailserver:/mail /mail nfs timeo=20, intr

Можно использовать команду **mount**:

mount –t nfs –o timeo=20, intr mailserver:/mail /mail

Поддержка файловых систем

Файловые системы полезно проверять на наличие поврежденных или разрушенных файлов. Для этого используется команда **fsck** (file system check):

fsck [-A] [-V] [-t fstype] [-a] [-l] [-r] [-s] filesystem

Чаще всего используется команда **fsck** filesystem.

-A	Проверяет все файловые системы, указанные в /etc/fstab
-V	Подробный отчет
-t fstype	Задает тип проверяемой файловой системы
Filesystem	Задает файловую систему. Можно указать как блочное

	устройство так и точку подключения
-a	Автоматически исправляет все ошибки
-l	Выводит список имен всех файлов
-r	Запрашивает подтверждение перед исправлением
-s	Выводит содержимое суперблока

Таблица 5 Опции fsck