6. РАСПРЕДЕЛЁННАЯ СЕТЕВАЯ ФАЙЛОВАЯ СИСТЕМА AFS

6.1 Общие сведения

AFS – это распределённая файловая система, способная объединять файловые системы множества файловых серверов, обладающая следующими преимуществами по сравнению с централизованными системами:

- *расширенная доступность*: копии часто используемых файлов (например, файлы приложений) могут храниться на множестве серверов, тогда при выходе из строя одной или даже нескольких машин такой файл может оказаться по-прежнему доступным, потому что запросы пользователей будут направлены на исправные машины;
- *улучшенная эффективность*: нагрузка может быть распределена между несколькими небольшими файловыми серверами, что зачастую оказывается выгоднее одного дорогого высокопроизводительного сервера.

При этом распределённость структуры AFS скрыта от пользователя. Работа с AFS мало чем отличается от работы с локальными файлами, хранящимися на машине пользователя. На машинах пользователя AFS представляется в виде иерархической файловой структуры, монтируемой, как правило, в каталог /afs.

В структуре AFS выделяют элементы, называемые *ячейками* (cell). Каждая ячейка представляет собой домен администрирования, в котором задаётся, как конфигурировать клиентские машины, какие пользователи имеют доступ к файловому пространству ячейки и т. п. Обычно в ячейку AFS объединяются компьютеры одной фирмы, факультета университета или определённой группы пользователей. Связанные между собой ячейки объединяются в AFS-*сайт* (site). Принято сопоставлять подкаталоги директории /afs файловым пространствам отдельных ячеек (например, каталоги и файлы ячейки фирмы Foo Inc. можно поместить в /afs/fooinc.com). Таким образом, хотя каждая ячейка AFS управляет и обслуживает собственное файловое пространство, существует возможность подключаться к файловым пространствам других ячеек. Та ячейка, к которой относится ваша клиентская машина, в файловом пространстве AFS называется *локальной ячейкой* (local cell), все остальные – *чужими ячейками* (foreign cells).

Обычно запоминающие устройства в компьютерах разделены на порции, называемые разделами. AFS производит дальнейшее деление предоставленного ей раздела на единицы, называемые *томами* (volumes). Том представляет собой удобный контейнер для хранения взаимосвязанных файлов и каталогов. Системный администратор может перемещать тома с одного сервера на другой, причём такое перемещение будет совершенно прозрачным для пользователя, т. к. AFS автоматически отслеживает место расположения тома. Пользователь получает доступ к содержимому тома через его точку монтирования в файловом пространстве AFS. Обычно для домашней директории каждого пользователя выделяется отдельный том, который монтируется в одну из папок локальной ячейки. Например, том пользователя pupkin в ячейке fooinc.com может быть смонтирован в каталог /afs/fooinc.com/home/pupkin. Для каждого тома администратор устанавливает предельный размер занимаемого им файлового пространства – *квоту*. При превышении этого предела возникает ошибка.

Для работы с AFS необходимо не только примонтировать файловое пространство AFS в выделенный каталог, но и запустить специальное клиентское программное обеспечение – *кэш-менеджер*. Когда пользователь обращается к файлу в AFS, кэш-менеджер определяет необходимый сервер и загружает с него копию этого файла на локальную машину. Прикладные программы используют локальную кэшированную копию файла, поэтому внесённые изменения не сразу отражаются на главной версии файла на сервере. Как правило, изменения переносятся, когда файл закрывается. Если во время работы с каким-либо файлом хранящий его сервер становится недоступным, можно продолжить работу с локальной копией, но изменения не будут сохранены до тех пор, пока сервер снова не станет доступным.

Когда главная копия файла на сервере изменяется, файловый сервер AFS сообщает всем кэш-менеджерам о недостоверности их локальных копий этого файла. Это происходит следующим образом. Файловый сервер вместе с копией файла передаёт кэш-менеджеру механизм обратного вызова (callback). При изменении главной копии файла файловый сервер разрушает обратный вызов. Когда программа запрашивает очередную порцию данных из изменённого файла, кэш-менеджер обнаруживает разорванный обратный вызов и загружает обновлённую копию файла. AFS сохраняет только изменения, сделанные в самую последнюю очередь. Поэтому необходимо предпринять определённые меры, чтобы не потерять свои изменения, если с файлом работают одновременно несколько пользователей. Один из возможных способов – воспользоваться средствами ограничения доступа AFS.

6.2 Аутентификация в AFS

Для обеспечения авторизованного доступа к файловому пространству ячейки используются пароли и взаимная аутентификация на основе протокола Kerberos. Во время аутентификации кэш-менеджер получает от сервера *токен* – порцию информации, зашифрованную при помощи вашего пароля. Если пользователь предоставил корректный пароль, кэш-менеджер может расшифровать токен. В дальнейшем токен служит

доказательством вашей аутентичности для серверных программ AFS. Более того, когда кэш-менеджер обращается к файловому серверу, он шифрует полученный пользователем токен при помощи ключа AFS-сервера, так что только истинный файловый сервер может его расшифровать, что обеспечивает взаимную аутентификацию.

Чтобы получить доступ к файловому пространству AFS, пользователь должен пройти процедуру регистрации на локальной машине и аутентифицировать себя в AFS. При использовании команды регистрации, модифицированной для AFS, эти две процедуры могут быть совмещены в одну. В этом случае регистрационная информация UNIX и AFS должна совпадать. Пользователь вводит в приглашении **login** своё имя и пароль и при успешной аутентификации автоматически получает AFS-токен. Проверить наличие токена можно при помощи команды **tokens**:

```
$ tokens
Tokens held by the Cache Manager:
User's (AFS ID 1022) tokens for afs@fooinc.com [Expires Feb 3 14:35]
--End of list--
```

В данном примере сообщается о наличии одного токена пользователя с идентификатором 1022 для ячейки fooinc.com, действительного до 3 февраля 14:35.

Если процесс регистрации в системе не подразумевает аутентификацию в AFS, необходимо сделать это явно при помощи команды **klog**. При успешной аутентификации данная программа передаёт кэш-менеджеру выделенный пользователю токен. Следует иметь ввиду, что можно получить только по одному токену для каждой ячейки AFS. Если вы запрашиваете ещё один токен для какой-то ячейки, то он заменит собой предыдущий выданный вам для этой ячейки токен. Более того, при создании токена для него задаётся определённое время жизни. По истечении этого времени токен перестаёт распознаваться как действительный, так что пользователю может быть отказано в доступе к AFS, поэтому необходимо вовремя запрашивать при помощи **klog** новый токен.

klog [**-setpag**] [**-cell** *ячейка*] [*пользователь*]

Если имя пользователя отсутствует, используется регистрационное имя в UNIX. Если не указывать ключ *-cell*, выдаётся токен для локальной ячейки. Ключ *-setpag* используется для привязки токена к определённой группе процессов (PAG – process authentication group), так что им могут воспользоваться только процессы из данного сеанса пользователя. В противном случае токен привязывается к UID пользователя, что позволяет воспользоваться токеном любому процессу с таким же UID.

Выход из системы не означает уничтожение токенов. Их следует удалять явно при помощи команды **unlog**.

unlog [**–cell** *ячейка*]

Если не указан ключ *–cell*, удаляются все токены, иначе удаляется только токен для указанной ячейки.

Для смены пароля в AFS используется команда **kpasswd**. Если используется модифицированная для AFS система регистрации, эта команда изменяет также системный пароль.

6.3 Информационные команды AFS

Большинство команд AFS выполняется при помощи сервиса **fs**:

fs *команда* [*параметры*]

Чтобы получить список доступных команд и краткое их описание, следует выполнить **fs help**. Подсказку по конкретной команде можно получить одним из следующих способов:

fs help *команда*
fs *команда* **-help**

Команда **fs quota** выводит информацию о проценте использованной квоты на томе, к которому относится указанный каталог (или текущий, если не указан). Более подробную информацию о томе можно получить при помощи команд **fs listquota** и **fs examine**.

```
$ fs quota /afs/fooinc.com/home/pupkin
   34% of quota used.
$ fs listquota /afs/fooinc.com/home/pupkin
   Volume Name     Quota    Used    % Used   Partition
   user.pupkin     10000    3400       34%         86%
$ fs examine /afs/fooinc.com/home/pupkin
   Volume status for vid = 536871122 named user.pupkin
   Current disk quota is 10000
   Current blocks used are 5745
   The partition has 1593 blocks available out of 99162
```

Приведённые команды сообщают имя тома, на котором расположен указанный каталог (или текущий, если не указан), величину квоты в Кбайтах, текущий размер файлового пространства в Кбайтах, а также информацию о разделе, на котором расположен этот том (общий размер и размер занятого файлового пространства).

Обычно пользователю нет необходимости знать, на каком именно файловом сервере AFS расположен тот или иной файл, тем не менее это можно определить при помощи команды **`fs whereis`**. Если файл вдруг станет недоступен, может оказаться полезным знать, где он на самом деле расположен, чтобы выяснить, всё ли в порядке с той машиной. Если в выводе команды упоминаются несколько серверов, значит, соответствующий том реплицирован на указанные сервера.

```
$ fs whereis /afs/fooinc.com/home/pupkin
File /afs/fooinc.com/home/pupkin is on host fs.fooinc.com
```

Иногда из-за программных или аппаратных проблем, а также во время технического обслуживания какие-либо сервера, составляющие ячейку AFS могут оказаться недоступны. Чтобы определить статус серверов в ячейке, используется команда:

`fs checkservers` [**`-cell`** *ячейка*] [**`-all`**]

Кэш-менеджер поддерживает список ячеек, к которым он может предоставить доступ. При помощи команды **`fs listcells`** можно получить этот список и имена серверов, составляющих ячейки.

6.4 Контроль доступа

Для управления доступом к файлам и папкам AFS использует *списки контроля доступа* (access control lists). Список контроля доступа (ACL) может содержать примерно до 20 элементов, разрешающих или запрещающих доступ для тех или иных пользователей или групп. Владелец каталога и системный администратор всегда имеют право на управление ACL данного каталога. Прочие пользователи могут менять ACL, только если для них в ACL специально определён такой тип доступа («*a*» – administer). Следует иметь в виду, что каждая ячейка AFS определяет свои группы и своих пользователей. Если пользователь принадлежит какой-либо группе, которая указана в ACL, пользователь получает все указанные типы доступа, как если бы он сам непосредственно был указан в ACL.

Все пользователи, которые имеют доступ к файловому пространству некоторой ячейки, вне зависимости от того, аутентифицированы они или нет, автоматически сопоставляются с предопределённой группой system:anyuser. Все аутентифицированные в данный момент в локальной ячейке пользователи сопоставляются в ней с предопределённой группой system:authuser.

AFS сопоставляет ACL каждому каталогу и применяет его ко всем файлам в данном каталоге. Обычные файлы не имеют отдельных ACL. При перемещении файла из одного каталога в другой изменяются атрибуты доступа к данному файлу. Таким образом, изменяя ACL каталога, пользователь изменяет атрибуты доступа ко всем файлам в данном каталоге. Создаваемый подкаталог наследует текущий ACL своего родительского каталога, но для этого подкаталога ACL может быть изменён позднее. Следует иметь в виду, что пользователь получит доступ к содержимому каталога, только если для этого каталога и всех его родительских каталогов установлен режим доступа ***l*** (lookup).

Таблица 6.1. Режимы доступа, определённые в AFS

Обозначение	Описание
l (lookup)	Право на поиск в каталоге. Позволяет пользователю просмотреть ACL данного каталога (**`fs listacl`**), информацию о данном каталоге (**`ls -ld`**), а также получить список файлов в данном каталоге (**`ls`**)
i (insert)	Право на добавление новых файлов и подкаталогов в данном каталоге. Не распространяется на подкаталоги, т. к. они имеют собственные ACL
d (delete)	Право на удаление файлов и подкаталогов в данном каталоге
a (administer)	Право на администрирование, т. е. на изменение ACL данного каталога
r (read)	Право на чтение файлов и просмотр информации о них
w (write)	Право на изменение самих файлов и изменение их UNIX-атрибутов доступа (**`chmod`**)
k (lock)	Право на блокирование файлов (на монопольный доступ)
all = rlidwka	Обозначает группу из всех семи режимов доступа
none	Используется для удаления записи из ACL, лишая пользователя или группу каких-либо полномочий
read = rl	Право на чтение файлов
write = rlidwk	Право на полный доступ к каталогу, за исключением изменения ACL

В команде изменения ACL (**`fs setacl`**) можно использовать комбинации режимов доступа, перечисленные в таблице 6.1, и псевдонимы для групп режимов доступа (*all*, *none*, *read*, *write*).

ACL может содержать *разрешающие элементы* (normal rights) и *запрещающие элементы* (negative rights). Запрещающие элементы имеют больший приоритет и используются для явного запрещения того или иного типа доступа для пользователя или группы.

AFS лишь частично использует атрибуты доступа UNIX, а именно AFS использует у обычных файлов биты, соответствующие классу доступа «владелец», игнорируя остальные биты прав доступа. Для каталогов AFS вообще не использует атрибуты доступа UNIX, контролируя доступ лишь с помощью ACL. Биты класса доступа «владелец» используются следующим образом:

- Если право «*r*» не установлено, никто (даже владелец) не имеет право читать файл вне зависимости от разрешений ACL. Если право «*r*» установлено, файл имеют право просматривать те, для кого в ACL заданы права «*r*» и «*l*».
- Если право «*w*» не установлено, никто (даже владелец) не имеет право изменять файл вне зависимости от разрешений ACL. Если право «*w*» установлено, файл может изменяться теми, для кого в ACL заданы права «*w*» и «*l*».
- Право «*x*» определяет, является ли файл исполняемым. Чтобы запустить файл на выполнение, пользователь должен иметь право читать его («*r*» и «*l*»).

Чтобы просмотреть ACL того или иного каталога, используется команда **`fs listacl`** (без параметров – ACL текущего каталога).

```
$ fs listacl /afs/fooinc.com/home/pupkin
   Access list for /afs/fooinc.com/home/pupkin is
   Normal rights:
      system:anyuser rl
      ivanov rlw
      petrov rlidwka
   Negative rights:
      ivanov:other-dept rl
      sidorov rl
```

В данном примере всем пользователям, кроме sidorov и тех, кто входит в группу ivanov:other-dept, разрешён доступ на чтение. Кроме того, пользователю ivanov разрешён доступ на запись, а пользователю petrov – полный доступ, если только эти пользователи не входят в группу ivanov:other-dept.

Для изменения ACL используется команда:

`fs setacl` [[**`-dir`**] *каталог*] [[**`-acl`**] *acl_spec*] \
[**`-negative`**] [**`-clear`**]

Если указана опция *-negative*, *acl_spec* задаёт запрещающие элементы. Если не указана – разрешающие. В качестве *acl_spec* указываются пары пользователь (группа) и атрибуты, разделённые пробелами. Внутри пары имя пользователя (группы) отделяется от атрибутов также пробелом. Если указана опция *-clear*, то перед установкой заданных элементов ACL будет очищен (все элементы удалены).

```
$ pwd
/afs/fooinc.com/home/pupkin
$ fs setacl sidorov none -negative
$ fs setacl -dir . -acl system:anyuser none \
   ivanov:colleagues write system:authuser rl
$ fs listacl
   Access list for /afs/fooinc.com/home/pupkin is
   Normal rights:
      system:authuser rl
      ivanov rlw
......ivanov:colleagues rlidwk
      petrov rlidwka
   Negative rights:
      ivanov:other-dept rl
```

В данном примере для каталога /afs/fooinc.com/home/pupkin удаляется запрещающий элемент ACL sidorov. Затем удаляется разрешающий элемент system:anyuser и добавляется доступ на запись для членов группы ivanov:colleagues, а также доступ на чтение для всех аутентифицированных пользователей (system:authuser).

Контрольные вопросы и задания

1. Аутентифицируйтесь в AFS. Выясните свой ID в AFS. Определите время жизни выданного вам токена (сколько часов?).
2. Определите название локальной ячейки AFS. Какие ячейки формируют доступное вам файловое пространство AFS?

3. Выясните название вашего персонального тома, его точку монтирования, квоту и файловый сервер, на котором он расположен.
4. Разрешите вашему соседу доступ в один из подкаталогов в вашей домашней папке на AFS-сервере. Убедитесь в возможности доступа. Отмените доступ.
5. Сравните системы контроля доступа AFS ACL и POSIX ACL по следующим критериям:
    1) возможность задать ACL для каждого файла,
    2) возможность явного запрещения доступа для пользователя или группы,
    3) возможность раздельного контроля доступа на добавление файлов и на удаление файлов в каталоге,
    4) кто имеет право изменять ACL,
    5) взаимодействие ACL со стандартными атрибутами доступа UNIX,
    6) как организовано наследование прав.