

Московский Государственный Технический Университет
имени Н. Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

УТВЕРЖДАЮ

Зав. кафедрой ИУ6

д.т.н., проф. _____ Сюзев В.В.

" ____ " _____ 2013 г.

Исследование методов организации внешней памяти.
Методические указания по выполнению лабораторной работы
по дисциплине "Операционные системы"

МОСКВА 2013

В настоящее время существует множество файловых систем различных категорий, например:

- Для носителей с произвольным доступом (FAT32, HPFS, ext4, NTFS);
- Для оптических носителей — CD и DVD: ISO9660, HFS, UDF;
- Виртуальные файловые системы: AEFS и др.;
- Сетевые файловые системы: NFS, CIFS, SSHFS, GmailFS и др.;
- Для флэш-памяти: YAFFS, ExtremeFFS, exFAT и др.

Цель работы: исследование файловых систем, применяющихся в UNIX-подобных системах, а также изучение основных утилит для работы с файлами.

Для выполнения данной ЛР рекомендуется использовать учетную запись суперпользователя (root), либо использовать su. Задания, выделенные курсивом, необязательны для выполнения.

Используемые термины

Терминология UNIX-систем отличается от терминологии Windows. Ниже приведены термины, которые будут использованы в данной ЛР.

Индексный дескриптор (inode) – структура данных, хранящая метаданные. Каждый индексный дескриптор имеет уникальный идентификатор в файловой системе.

Жёсткая ссылка (hard link, hardlink) – составляющая каталога, ассоциирует имя файла с содержимым на диске. Прописывается в индексном дескрипторе файла.

Символьная ссылка (символическая ссылка, мягкая ссылка, symbolic link, symlink) – представляет собой структуру данных, которая содержит в себе только ссылку на другой файл. Символьные ссылки часто используются как псевдонимы. В Windows тоже существуют символьные ссылки. Доступ к символьной ссылке не ограничен, права доступа проверяются при попытке доступа к целевому файлу.

Архиватор – утилита для создания архивов. В отличие от Windows, где под архивом понимается файл, содержащий сжатые данные, в UNIX-подобных системах архив может быть и не сжатым, это просто информация

для длительного хранения, обращение к которой происходит достаточно редко. Утилиты для сжатия файлов называются компрессорами.

Следует отметить, что все устройства в UNIX являются файлами. Все они монтируются в /dev. Примерами могут являться /dev/sda (первый жёсткий диск со SCSI-интерфейсом), /dev/hda (жёсткий диск с IDE-интерфейсом), /dev/null (специальное устройство, в него можно записать информацию любого объёма, при чтении выдаёт EOF, часто используется для перенаправления ненужных сообщений).

Специальными типами файлов являются также каналы (pipes) и порты (sockets, в некоторой литературе - гнёзда).

Имена папок в UNIX тоже имеют смысл. Например, /etc – директория, содержащая конфигурационные файлы (сокращение от et cetera – «и так далее»), /bin содержит исполняемые файлы (binary), /home – домашние директории пользователей, /lib – библиотеки (libraries) и т.д.

Каждая директория имеет минимум две ссылки: '.' – ссылка на саму же директорию, и '..' – ссылка на родительскую директорию. Эти ссылки являются жёсткими.

Основные команды, используемые при работе с ФС

В данном разделе приводятся утилиты, использующиеся при работе с файлами и файловыми системами. Некоторые из них были использованы в предыдущих ЛР.

`mount` – подключение (монтирование) файловой системы.

`umount` – отключение файловой системы.

`mkfs` – создание новой файловой системы. Существуют специализированные утилиты для создания ФС определённого типа (например, `mkfs.ext4`, `mkfs.ntfs`, полный список на рисунке 1).



```
root@debian:~# mkfs
mkfs          mkfs.ext2      mkfs.ext4dev  mkfs.ntfs
mkfs.bfs      mkfs.ext3      mkfs.minix    mkfs.vfat
mkfs.jramfs   mkfs.ext4      mkfs.msdos
```

Рисунок 1 – Утилиты mkfs

`mke2fs` – создание ext2/3/4 ФС.

`fdisk` – утилита для управления разделами диска.

`df` – отображает информацию об использовании диска.

`du` – отображает информацию об использовании диска. Разница показана на рисунке 2. Более подробную информацию можно узнать в `man`.

```
root@debian:~# df
Файловая система          1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
rootfs                    7867856      2274344    5193848         31% /
udev                      10240         0         10240          0% /dev
tmpfs                     38476         260         38216          1% /run
/dev/disk/by-uuid/8ed2d8a1-6e37-4952-86b2-0901cd3d66f1 7867856      2274344    5193848         31% /
tmpfs                      5120         0          5120          0% /run/lock
tmpfs                     155380         0         155380          0% /run/shm

root@debian:~# du
4      ./thumbnails/normal/00000000000000000000000000000000.png
8      ./thumbnails/normal
4      ./thumbnails/large/00000000000000000000000000000000.png
8      ./thumbnails/large
20     ./thumbnails
0      ./gvfs
8      ./dbus/session-bus
12     ./dbus
12     ./ideesktop
8      ./config/pcmanfm/default
12     ./config/pcmanfm
4      ./config/abiword/templates
32     ./config/abiword
8      ./config/tint2
4      ./config/lxpanel/default
8      ./config/lxpanel
32     ./config/openbox
8      ./config/leafpad
8      ./config/xfce4/panel/launcher-11
8      ./config/xfce4/panel/launcher-9
8      ./config/xfce4/panel/launcher-10
28     ./config/xfce4/panel
8      ./config/xfce4/xfconf/xfce-perchannel-xml
12     ./config/xfce4/xfconf
48     ./config/xfce4
8      ./config/libfm
164    ./config
36     ./fluxbox
8      ./local/share/applications
12     ./local/share
16     ./local
4      ./aptitude
12     ./cache/menus
4      ./cache/openbox/sessions
12     ./cache/openbox
28     ./cache
16     ./w3m
504    .
root@debian:~#
```

Рисунок 2 – Вывод утилит `df` и `du`

`ls` – вывод списка директорий (аналогичные утилиты – `dir`, `vdir`).

`cp` – копирование файла.

`mv` – перемещение файла (альтернативное применение – переименование).

`rm` – удаление.

`cd` – смена директории.

`dd` – утилита для копирования и конвертации файлов.

`ln` – создание ссылки.

`pwd` – вывод рабочего каталога.

`mkdir` – создание папки.

`rmdir` – удалить пустую папку.

`basename` – убирает из имени файла имена каталогов и суффиксы (преобразование полного пути в имя файла).

`dirname` – преобразует полный путь к файлу в имя родительской директории.

`cat` – просмотр файла.

`head` – вывод первых строк файла.

`tail` – вывод последних строк файла.

`less` – текстовый редактор.

`od` – просмотр в восьмеричном, десятичном или шестнадцатеричном формате. Пример работы показан на рисунке 3.

```
root@debian:~# od /etc/xdg/openbox/autostart
0000000 005043 020043 064124 071545 020145 064164 067151 071547
0000020 060440 062562 071040 067165 073440 062550 020156 067141
0000040 047440 062560 061156 074157 054040 051440 071545 064563
0000060 067157 064440 020163 072163 071141 062564 027144 021412
0000100 054440 072557 066440 074541 070040 060554 062543 060440
0000120 071440 066551 066151 071141 071440 071143 070151 020164
0000140 067151 022040 047510 042515 027057 067543 063156 063551
0000160 067457 062560 061156 074157 060457 072165 071557 060564
0000200 072162 021412 072040 020157 072562 020156 071565 071145
0000220 071455 062560 064543 064546 020143 064164 067151 071547
0000240 005056 005043 021412 044440 020146 067571 020165 060567
0000260 072156 072040 020157 071565 020145 047107 046517 020105
0000300 067543 063156 063551 072040 067557 071554 027056 005056
0000320 005043 064443 020146 062564 072163 026440 020170 072457
0000340 071163 066057 061151 067457 062560 061156 074157 063457
0000360 067556 062555 071455 072145 064564 063556 026563 060544
0000400 066545 067157 037040 062057 073145 067057 066165 035554
0000420 072040 062550 005156 020043 027440 071565 027562 064554
0000440 027542 070157 067145 067542 027570 067147 066557 026545
0000460 062563 072164 067151 071547 062055 062541 067555 020156
0000500 005046 062443 064554 020146 064167 061551 020150 067147
0000520 066557 026545 062563 072164 067151 071547 062055 062541
0000540 067555 020156 027476 062544 027566 072556 066154 020073
0000560 064164 067145 021412 020040 067147 066557 026545 062563
0000600 072164 067151 071547 062055 062541 067555 020156 005046
0000620 063043 005151 021412 044440 020146 067571 020165 060567
0000640 072156 072040 020157 071565 020145 043130 042503 061440
0000660 067157 064546 020147 067564 066157 027163 027056 021412
0000700 021412 063170 062543 066455 071543 066455 067141 063541
0000720 071145 023040 074012 061546 032145 070055 067141 066145
0000740 023040 064412 062544 065563
0000750
root@debian:~#
```

Рисунок 3 – Утилита `od`

xxd – просмотр файлов в шестнадцатеричном или двоичном формате. Сравнение с od показано на рисунке 4.

```
00000b0: 6e74 2074 6f20 7573 6520 474e 4f4d 4520 nt to use GNOME
00000c0: 636f 6e66 6967 2074 6f6f 6c73 2e2e 2e0a config tools....
00000d0: 230a 2369 6620 7465 7374 202d 7820 2f75 #.##if test -x /u
00000e0: 7372 2f6c 6962 2f6f 7065 6e62 6f78 2f67 sr/lib/openbox/g
00000f0: 6e6f 6d65 2d73 6574 7469 6e67 732d 6461 nome-settings-da
0000100: 656d 6f6e 203e 2f64 6576 2f6e 756c 6c3b emon >/dev/null;
0000110: 2074 6865 6e0a 2320 202f 7573 722f 6c69 then.# /usr/li
0000120: 622f 6f70 656e 626f 782f 676e 6f6d 652d b/openbox/gnome-
0000130: 7365 7474 696e 6773 2d64 6165 6d6f 6e20 settings-daemon
0000140: 260a 2365 6c69 6620 7768 6963 6820 676e &.#elif which gn
0000150: 6f6d 652d 7365 7474 696e 6773 2d64 6165 ome-settings-dae
0000160: 6d6f 6e20 3e2f 6465 762f 6e75 6c6c 3b20 mon >/dev/null;
0000170: 7468 656e 0a23 2020 676e 6f6d 652d 7365 then.# gnome-se
0000180: 7474 696e 6773 2d64 6165 6d6f 6e20 260a ttings-daemon &.
0000190: 2366 690a 0a23 2049 6620 796f 7520 7761 #fi...# If you wa
00001a0: 6e74 2074 6f20 7573 6520 5846 4345 2063 nt to use XFCE c
00001b0: 6f6e 6669 6720 746f 6f6c 732e 2e2e 0a23 onfig tools....#
00001c0: 0a23 7866 6365 2d6d 6373 2d6d 616e 6167 .#xfce-mcs-manag
00001d0: 6572 2026 0a78 6663 6534 2d70 616e 656c er &..xfce4-panel
00001e0: 2026 0a69 6465 736b &..idesk
root@debian:~# od -x /etc/xdg/openbox/autostart
0000000 0a23 2023 6854 7365 2065 6874 6e69 7367
0000020 6120 6572 7220 6e75 7720 6568 206e 6e61
0000040 4f20 6570 626e 786f 5820 5320 7365 6973
0000060 6e6f 6920 2073 7473 7261 6574 2e64 230a
0000100 5920 756f 6d20 7961 7020 616c 6563 6120
0000120 7320 6d69 6c69 7261 7320 7263 7069 2074
0000140 6e69 2420 4f48 454d 2e2f 6f63 666e 6769
0000160 6f2f 6570 626e 786f 612f 7475 736f 6174
0000200 7472 230a 7420 206f 7572 206e 7375 7265
0000220 732d 6570 6963 6966 2063 6874 6e69 7367
0000240 0a2e 0a23 230a 4920 2066 6f79 2075 6177
0000260 746e 7420 206f 7375 2065 4e47 4d4f 2045
0000300 6f63 666e 6769 7420 6f6f 736c 2e2e 0a2e
0000320 0a23 6923 2066 6574 7473 2d20 207f 752f
0000340 7273 6c2f 6269 6f2f 6570 626e 786f 672f
0000360 6f6e 656d 732d 7465 6974 676e 2d73 6164
0000400 6d65 6e6f 3e20 642f 7665 6e2f 6c75 3b6c
0000420 7420 6568 0a6e 2023 2f20 7375 2f72 696c
0000440 2f62 706f 6e65 6f62 2f78 6e67 6d6f 2d65
0000460 6573 7474 6e69 7367 642d 6561 6f6d 206e
0000500 0a26 6523 696c 2066 6877 6369 2068 6e67
0000520 6d6f 2d65 6573 7474 6e69 7367 642d 6561
0000540 6f6d 206e 2f3e 6564 2f76 756e 6c6c 203b
0000560 6874 6e65 230a 2020 6e67 6d6f 2d65 6573
0000600 7474 6e69 7367 642d 6561 6f6d 206e 0a26
0000620 6623 0a69 230a 4920 2066 6f79 2075 6177
0000640 746e 7420 206f 7375 2065 4e58 4543 6320
0000660 6e6f 6966 2067 6f74 6c6f 2e73 2e2e 230a
0000700 230a 6678 6563 6d2d 7363 6d2d 6e61 6761
0000720 7265 2620 780a 6366 3465 702d 6e61 6c65
0000740 2620 690a 6564 6b73
0000750
root@debian:~#
```

Рисунок 4 – Вывод утилит xxd и od

touch – создание пустого файла.

umask – задание маски прав доступа для новых файлов и директорий.

stat – вывод атрибутов файлов и директорий.

file – определение типа файла.

`size` – определение размера файла.

`chown` – изменение владельца или группы для файла или директории.

`chgrp` – изменение группы файла или директории. Может быть исполнена обычным пользователем, но в таком случае могут быть изменены только те группы, членами которых является пользователь.

`chmod` – смена прав доступа.

`chattr` – смена атрибутов файла в ext2.

`cksum` – расчет количества байт и контрольной суммы файла.

`fuser` – идентификация процессов, использующих указанные файлы или порты.

`find` – утилита поиска.

`whereis` – поиск файлов определённого приложения. Примеры использования и сравнение показаны на рисунке 5.

```
root@debian:~# find /bin/ -name 'login'
/bin/login
root@debian:~# whereis login
login: /bin/login /etc/login.defs /usr/share/man/man1/login.1.gz /usr/share/man/man3/login.3.gz
```

Рисунок 5 – Утилиты `find` и `whereis`

`which` – отображает полный путь к программе или скрипту. Пример показан на рисунке 6.

```
root@debian:~# which which
/usr/bin/which
```

Рисунок 6 – Утилита `which`

`cmp` – утилита сравнения, может быть использована для файлов.

`diff` – утилита сравнения файлов, выводящая разницу. Утилита `diff3` сравнивает 3 файла.

`cpio` – архиватор.

`tar` – компрессор `tar`.

`xz` – компрессор `xz`, на рисунке 7 показаны все утилиты этого пакета.

```
root@debian:~# xz
xz      xzcat    xzcmp    xzdiff  xzegrep  xzfgrep  xzgrep  xzless  xzmore
```

Рисунок 7 – Утилиты для работы с форматом хз

bzip2 – компрессор.

gzip – компрессор.

mkswap – создание файла подкачки (swap).

mktemp – создание временного файла.

Файловая система ext4

Данная секция описывает некоторые особенности файловой системы ext4 и содержит в себе практическую часть.

Некоторые особенности ext4fs:

- Экстенты (extents) – вместо поблочной адресации допустимо адресовать до 128 Мб идущих подряд блоков одним дескриптором.
- Контрольная сумма в журнале – в отличие от ext2, ext3 и ext4 являются журналируемыми файловыми системами, в ext4 хранится контрольная сумма транзакций диска.
- Мультиблочное выделение (выделение блоков группами, multiblock allocation) – файловая система хранит данные не только об отдельных свободных блоках, но и о группах свободных блоков. При записи файла производится поиск группы блоков, на которые данные могут быть записаны без фрагментации.
- Отложенное выделение (delayed allocation) – выделение блоков происходит перед физической записью на диск (а не при системном вызове write).
- Предварительное выделение (persistent pre-allocation) – возможно заранее зарезервировать место для записи файла.
- Версия inode (индексного дескриптора) – у индексного дескриптора появился номер, увеличивающийся каждый раз при изменении дескриптора.
- Обратная совместимость – файловые системы ext2 и ext3 могут быть монтированы как ext4.

Файловая система ext4 используется по умолчанию в дистрибутив Debian 7.2, который изучается в качестве примера в данной серии ЛР.

b – блочное устройство;
c – символьное устройство;
p – канал (pipe);
l – символическая ссылка;
s – порт (socket).

-d – обрабатывать каталог как файл.
-r – сортировка в обратном порядке.
-t – сортировать по штампу времени.

Примеры работы утилиты ls с некоторыми из этих опций показаны на рисунке 10.

```
root@debian:~# ls -iR /etc/xdg/xfce4/
/etc/xdg/xfce4/:
6366 helpers.rc  6373 panel  6370 xfconf

/etc/xdg/xfce4/panel:
6374 default.xml

/etc/xdg/xfce4/xfconf:
6371 xfce-perchannel-xml

/etc/xdg/xfce4/xfconf/xfce-perchannel-xml:
6372 xfce4-keyboard-shortcuts.xml
root@debian:~# ls -iRt /etc/xdg/xfce4/
/etc/xdg/xfce4/:
6373 panel  6370 xfconf  6366 helpers.rc

/etc/xdg/xfce4/panel:
6374 default.xml

/etc/xdg/xfce4/xfconf:
6371 xfce-perchannel-xml

/etc/xdg/xfce4/xfconf/xfce-perchannel-xml:
6372 xfce4-keyboard-shortcuts.xml
root@debian:~# ls -l /etc/xdg/xfce4/helpers.rc
-rw-r--r-- 1 root root 223 Июн 30 2012 /etc/xdg/xfce4/helpers.rc
root@debian:~#
```

Рисунок 10 – Вывод утилиты ls с различными аргументами

В качестве дополнительного задания покажите работу (с различными опциями) следующих утилит: ctp, diff, fuser.

Утилита ln позволяет создавать как жёсткие, так и символические ссылки. Создание жёсткой ссылки на файл:

ln file hardlink

С утилитой ln используются следующие ключи:

- s – создать символическую ссылку.
- d – сделать жёсткую ссылку для директории (только для суперпользователя).
- b – делать резервные копии файлов, которые будут удалены или изменены.

Пример создания символической ссылки показан на рисунке 11.

```
root@debian:~# ln -s /home/user/file /root/symlink
root@debian:~# ls -l /root/symlink
lrwxrwxrwx 1 root root 15 дек 14 03:28 /root/symlink -> /home/user/file
root@debian:~#
```

Рисунок 11 – Работа с символической ссылкой

Также значения ссылок можно считывать с помощью утилиты readlink:

```
readlink /root/symlink
```

Практическое задание:

1. Создайте пустой файл (с помощью touch).
2. Отобразите о нём информацию (с помощью ls).
3. Сделайте на него жёсткую и символическую ссылки. Посмотрите, как изменится информация о файле.
4. Внесите изменения в созданный файл (например, с помощью leafpad) и сохраните их. Откройте символическую ссылку с помощью leafpad. Сделайте то же самое с жёсткой ссылкой.

Ко всем пунктам задания должны быть приложены скриншоты.

Файловая система NFS

NFS (Network File System), по сути, представляет собой не самостоятельную файловую систему, а протокол для сетевого доступа к файловым системам. NFS абстрагируется от файловых систем сервера и клиента, можно представить, что NFS – это файловая система над файловыми системами.

Сервер NFS экспортирует свои каталоги, которые будут использоваться клиентами (обычно их список хранится в /etc/exports). Клиенты монтируют эти каталоги, монтируемые каталоги становятся частью иерархии файловой

системы клиентов, хотя на самом деле они остаются на сервере и доступ к ним осуществляется с помощью механизма удалённого вызова процедур. На рисунке 12 показан пример монтирования удалённых файловых систем из книги Э.Таненбаума «Современные операционные системы», каталоги показаны в виде квадратов, файлы – в виде кружков.

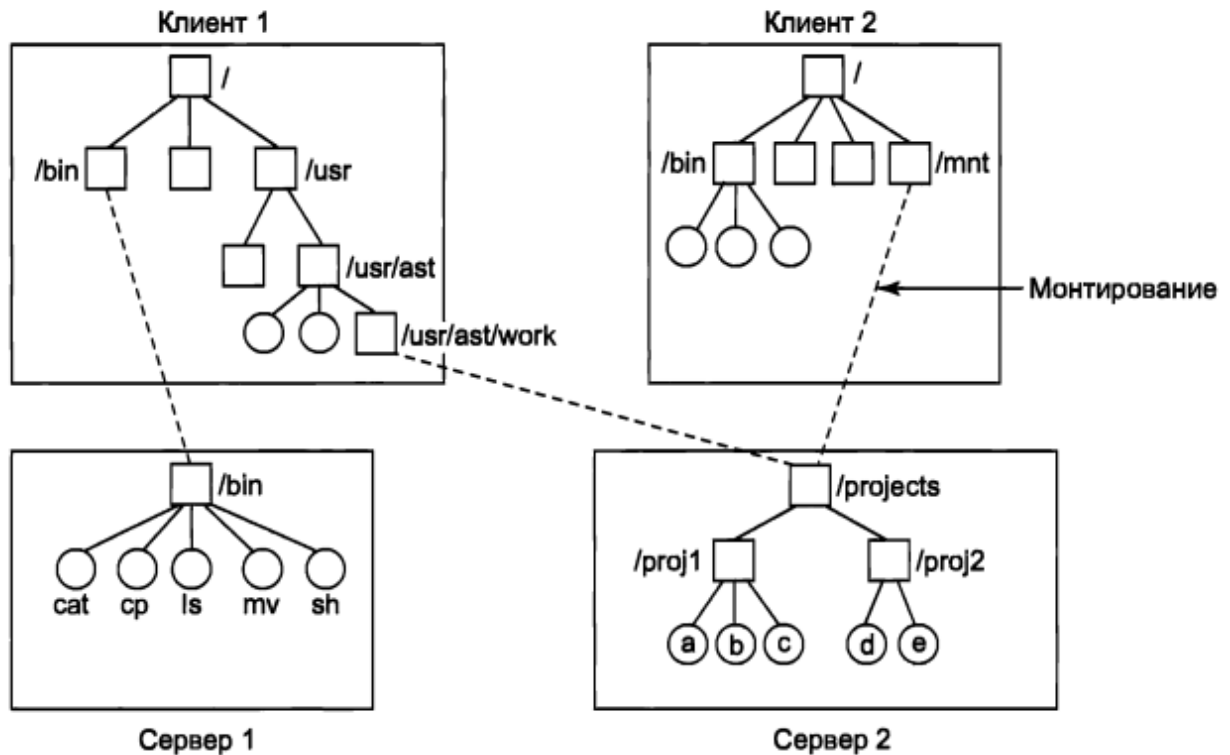


Рисунок 12 – Монтирование удалённых каталогов

NFS в Linux реализована с помощью нескольких уровней. Верхний уровень – уровень системных вызовов, он алогичен таковому для локальных ФС (обработка системных вызовов открытия файла, записи и т.д.). После обработки системный вызов передаётся на уровень виртуальной файловой системы (VFS). Виртуальная файловая система обслуживает таблицы, содержащие записи виртуальных индексных дескрипторов (virtual inode, v-узел) для всех открытых файлов. Нижний уровень – локальные файловые системы. Пример организации NFS из книги «Современные операционные системы» показан на рисунке 13.

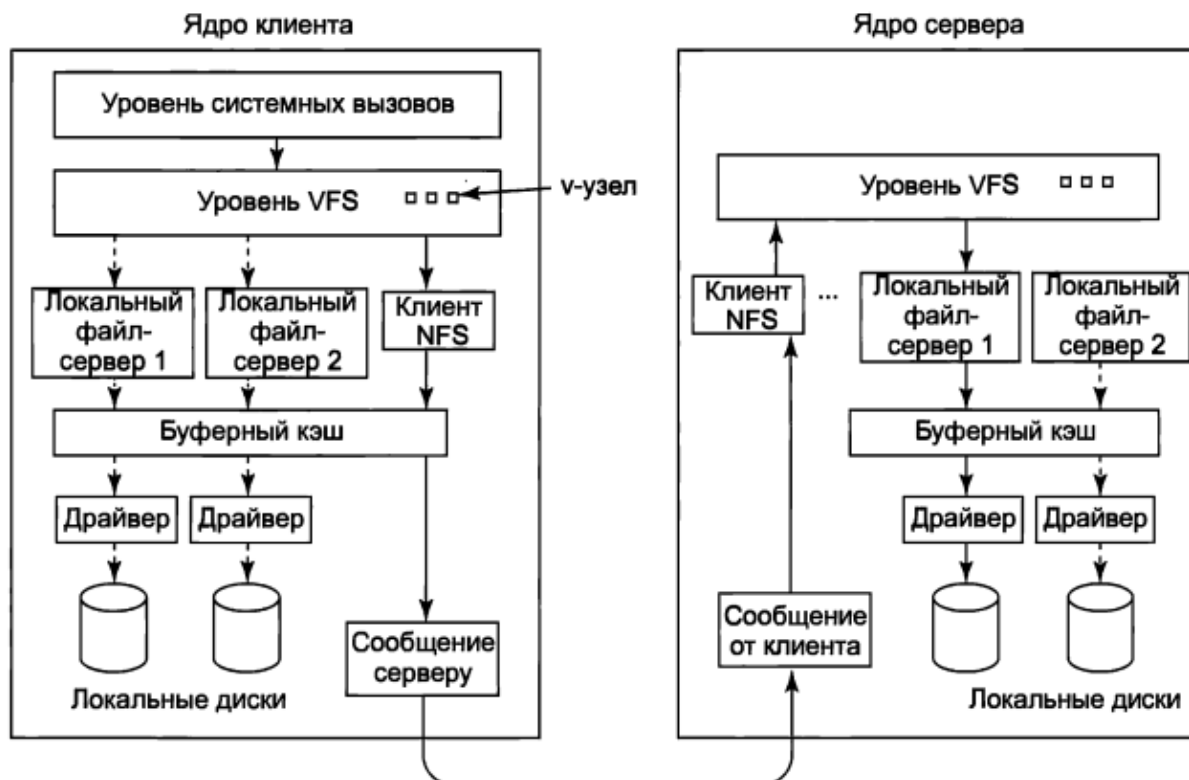


Рисунок 13 – Структура уровней файловой системы NFS

На сегодняшний день актуальной версией NFS является NFS v4. В отличие от своих предшественниц, 4 версия является ФС с сохранением состояния. Это стало возможным благодаря введению составных (compound) операций: вместо одной большой транзакции передавалось множество транзакций с малым объёмом данных.

В качестве практики будет показан краткий пример настройки NFS-сервера.

1. Выбор экспортируемых папок и настройка сервера.

Отредактируйте файл `/etc/exports` таким образом, чтобы он содержал записи следующего вида:

```
directory/FS client1 (options) client2 (options) ...
```

Некоторые общие опции:

`rw` – доступ для чтения и записи (по умолчанию – только чтение, `ro`);

`async` – использовать асинхронный метод обработки (по умолчанию – синхронный, `sync`);

Некоторые опции отображения пользователей:

`root_squash` – не позволять пользователю `root` обращаться к смонтированному тому (противоположность – `no_root_squash`);

`all_squash` – вместо UID и GID используется запись анонимного пользователя, используется для открытых томов (по умолчанию – `no_all_squash`);

`anonuid/anongid` – меняют данные учётной записи анонимного пользователя на указанные.

Пример файла конфигурации показан на рисунке 14.

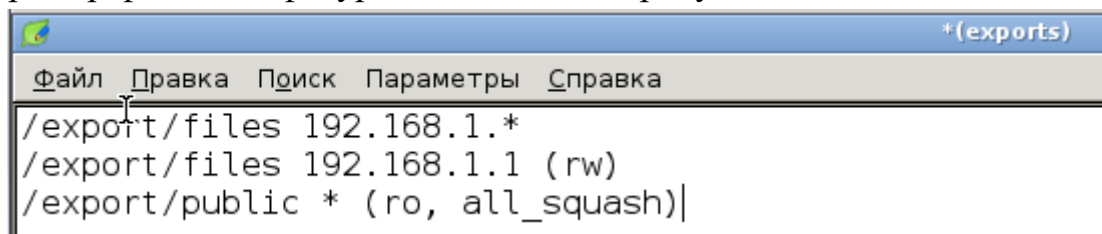


Рисунок 14 – Настройка доступа к экспортируемым каталогам

На рисунке записи означают следующее:

Предоставить доступ всем компьютерам в сети 192.168.1. Все настройки – по умолчанию.

Предоставить клиенту 192.168.1.1 права на чтение и изменение каталога.

Предоставить всем клиентам доступ к `/export/public`, но только для чтения и только под учётной записью анонимного пользователя.

2. Настройка клиента.

Клиент должен монтировать экспортируемую сервером файловую систему командой следующего вида:

```
mount server:directory local_mount_point
```

Например, сервер имеет IP 192.168.0.1, точка монтирования - `/mnt`:

```
mount 192.168.0.1:/export/files /mnt
```

Файловая система GPFS

GPFS (General Parallel File System) – параллельная кластерная файловая система от IBM, впервые представлена IBM в 1998 году. Используется в кластерах (часто в IBM x), при организации облачных инфраструктур, совместно с суперкомпьютерами (IBM Blue Gene).

Файл в GPFS может быть распределён по нескольким дискам, находящимся на различных узлах, при этом для работы с файловой системой могут использоваться обычные утилиты UNIX, как в случае с локальной ФС.

Преимущества GPFS:

- Высокая производительность – ресурсы GPFS доступны одновременно многим процессорам на многих узлах.
- Высокий показатель восстанавливаемости и высокая доступность – GPFS является журналируемой файловой системой, причём для

каждого узла ведётся собственный журнал. GPFS включает в себя утилиты для гибкой настройки и администрирования.

- Гибкость – узлы могут добавляться и удаляться без перемонтирования файловой системы.

Рисунок 15 показывает компоненты GPFS и их взаимодействие.

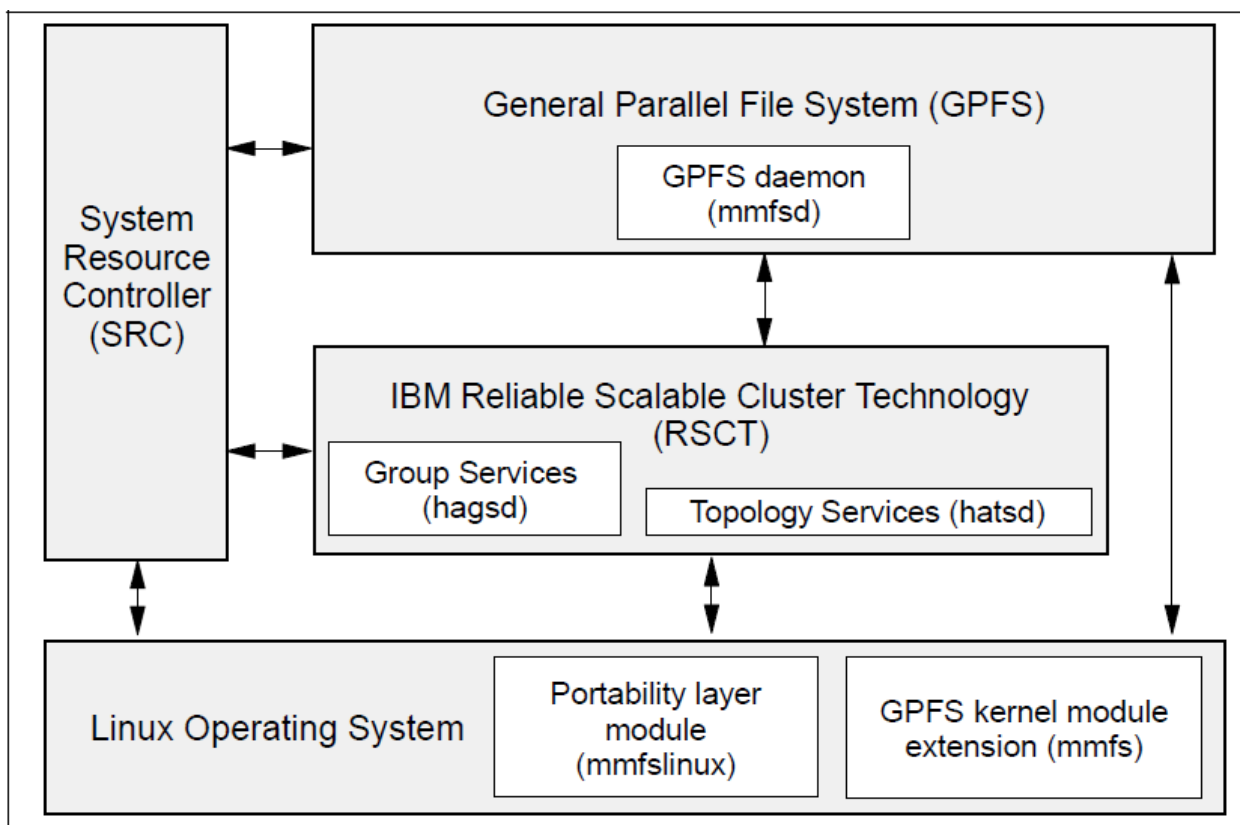


Рисунок 15 – Компоненты GPFS

IBM также отмечает, что, в отличие от NFS, GPFS является самостоятельной файловой системой, что положительно сказывается на быстродействии и удобстве администрирования.

В данной лабораторной работе будет использоваться кластер МГТУ «Королёв» (IBM System Cluster 1350 “Korolyov”). Логин и пароль для доступа можно получить у преподавателя. На рисунке 16 показан пример входа в систему с помощью утилиты ssh, изучите его и повторите с полученным логином и паролем. Соединение может занять некоторое время.

```

root@debian:~# ssh stud160@hpc.bmstu.ru -p2022
The authenticity of host '[hpc.bmstu.ru]:2022 ([195.19.33.110]:2022)' can't be established.
RSA key fingerprint is 20:ef:d0:9f:ee:ed:9d:43:15:ea:03:06:9f:3a:40:dd.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '[hpc.bmstu.ru]:2022,[195.19.33.110]:2022' (RSA) to the list of known hosts.
--
Currently powered on following nodes:
    n1101 ... n1113
    n1201 ... n1213 (except n1204)
--
stud160@hpc.bmstu.ru's password:

```

Рисунок 16 – Диалог авторизации ssh

Проверьте имя своей домашней директории (введя \$HOME).

Помимо стандартных утилит UNIX, GPFS использует свои специальные утилиты. Ниже перечислены некоторые из них:

`mmgetacl` – получить список контроля доступа (ACL, Access Control List).

`mmputacl` – установить список контроля для файла.

`mmeditacl` – изменить список контроля для файла.

`mmdeacl` – удалить список контроля.

`mmclscluster` – показать информацию о кластере GPFS (недоступно для обычных пользователей).

`mmclsmgr` – отобразить управляющие узлы для различных ФС.

`mmclsnsd` – отобразить разделяемые диски NSD (Network Shared Disk), недоступно для пользователя, пример использования показан на рисунке 17.

File system	Disk name	NSD servers
gpfs	lun0	gpfs1.nodes, gpfs2.nodes

Рисунок 17 – Вывод утилиты `mmclsnsd`

Списки контроля доступа представляют собой обычные текстовые файлы, формат похож на формат списков доступа утилит `getfacl/setfacl`, рассмотренных в предыдущей ЛР. Пример такого списка из официальной документации показан на рисунке 18.


```
user::rwx
group::rwx
other::--x
mask::rw-
user:shill:rwx
group:itso:rwx
group:control:-w-
```

Рисунок 18 – Пример ACL в GPFS

Пример установки списка контроля доступа:

1. Создание списка ACL:

```
cat > acl
```

В файле набрать данные, показанные на рисунке 18, обратите внимание, что используется 4 бита для каждого поля. Для окончания редактирования нажать Ctrl-D. Проверить данные можно командой `cat acl`.

```
[18:24 stud160@mgmt ~]$cat > acl
user::rwx
group::rwx
other::rwx
[18:25 stud160@mgmt ~]$cat acl
user::rwx
group::rwx
other::rwx
```

Рисунок 18 – Создание списка контроля доступа.

2. Создать файл с помощью touch:

```
touch $HOME/file
```

3. Получить список контроля доступа для файла (рисунок 19).

```
[18:26 stud160@mgmt ~]$mmgetacl file
#owner:stud160
#group:iu6
user::rw-
group::r---
other::r---
[18:26 stud160@mgmt ~]$
```

Рисунок 19 – Список контроля доступа созданного файла

4. Установить созданный список доступа проверить (рисунок 20).

```
[18:26 stud160@mgmt ~]$mmputacl -i acl file
[18:27 stud160@mgmt ~]$mmgetacl file
#owner:stud160
#group:iu6
user::rwx
group::rwx
other::rwx
[18:27 stud160@mgmt ~]$
```

Рисунок 20 – Установка и проверка списка контроля для файла

Данные об операциях хранятся в файлах mmfs.log (ознакомьтесь с содержимым файла /var/adm/ras/mmfs.log.latest).

Практическое задание:

1. Создать свой файл в домашней директории.
2. Проверить его ACL несколькими способами.
3. Создать новый ACL.
4. Установить новый ACL, показать это выводом соответствующей команды.
5. Удалить созданные файлы.

Ко всем пунктам задания привести скриншоты.

Завершить сессию ssh можно нажатием Ctrl-D (в консоли должна появиться надпись «connection to hpc.bmstu.ru closed»).

Выводы

В данной ЛР были рассмотрены некоторые из применяющихся на данный момент в UNIX-подобных системах файловые системы. Также были рассмотрены основные утилиты для работы с файлами.

Файловая система ext4 представляет собой развитие файловой системы ext3. Она может содержать большее количество каталогов, адресовать большие диски, эффективнее использует пространство при размещении больших файлов. ФС ext4 используется по умолчанию в ряде дистрибутивов, в том числе и Debian (начиная с версии 7.0).

Файловая система NFS основана на удалённом вызове процедур (RPC) и предоставляет протокол для взаимодействия с удалёнными ресурсами. NFS не является самостоятельной ФС, она позволяет монтировать папки из различных файловых систем.

GPFS является высокопроизводительной параллельной файловой системой. В отличие от NFS, GPFS является самостоятельной ФС. Эта ФС является проприетарной. Она может использоваться как в системах высокой доступности, так и в высокопроизводительных системах.

Контрольные вопросы:

1. Ярлык и символическая ссылка – одно и то же?
2. Зачем используется umask? В чём отличие от chmod?
3. Зачем может использоваться версия inode?
4. В чём достоинство NFS?

Порядок выполнения работы.

1. Ознакомиться с теоретическими сведениями по файловым системам.
2. Изучить и использовать утилиты и команды для работы с файлами.
3. Пройти тестирование по теоретическому материалу.

Отчет должен включать:

- название работы и ее цель;
- описание команд и результаты работы.