Факультет
биоинженерии и биоинформатики
Московский государственный университет имени М.В.Ломоносова

Базовое устройство GNU/Linux

Операционная система

Структура GNU/Linux

Пользовательское пространство (user space)

Пользовательские приложения (applications)

Системные утилиты, демоны, библиотеки
(utilities, daemons, libraries)

Стандартная библиотека языка Си (libc)

Системные вызовы (syscalls)

Ядро Linux (kernel)

Загружаемые модули ядра (LKM)

Пространство ядра (kernel space)

Функции ядра ОС

- распределение ресурсов между процессами
- изоляция процессов и обмен информацией между процессами
- контроль доступа и обеспечение безопасности
- абстракция оборудования

Ядро Linux

- монолитное
- модульное с поддержкой загружаемых модулей
- многозадачное
- с поддержкой виртуальной памяти
- написано на Си и Rust

Стандартная библиотека Си

Включает:

- стандартные функции Си (например, C11)
- функции стандарта POSIX
- Linux-специфические функции (обертки системных вызовов)

Реализации:

glibc – GNU C library, основная реализация в Linux

klibc – урезанная версия, используется при загрузке ядра

musl – альтернативная реализация, с акцентом на эффективном статическом связывании

uClibc – для встраиваемых систем

Bionic – реализация от Google для Android

Файлы

Виды файловых систем

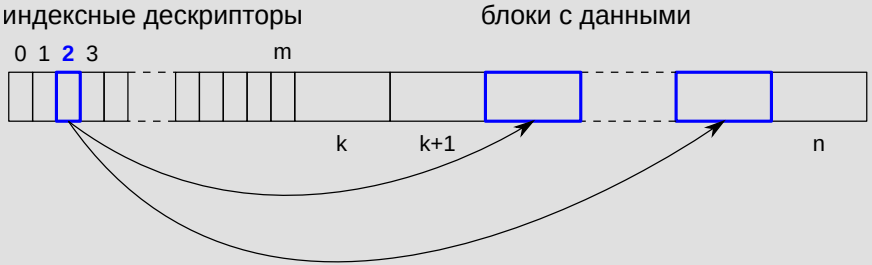
- дисковые (в широком смысле)
  - ext4, ext3, ext2
  - FAT16, FAT32
  - NTFS, APFS, HFS+, UFS
  - XFS, Btrfs, ReiserFS
  - ISO 9660, UDF
  - …
- сетевые
  - NFS
  - CIFS (SMB)
  - SSHFS
  - …
- специализированные
  - tmpfs
  - procfs, sysfs, devfs, …
  - FUSE
  - …

Виртуальная файловая система

Представление любой файловой системы с точки зрения ядра Linux.

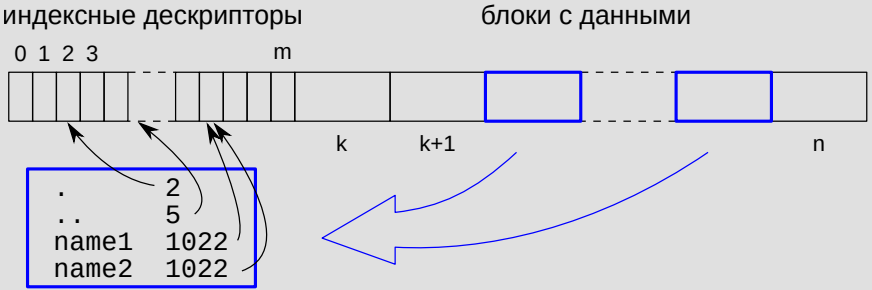
Inode

Любому объекту файловой системы соответствует индексный дескриптор (inode).



Директории

Папка – особый вид файлов (в широком смысле).



Специальные файлы

Файлы в широком смысле (объекты файловой системы):

- `-` обычные (regular) файлы
- `d` папки (каталоги, директории)
- `l` символические ссылки
- `s` сокеты
- `b` файлы блочных устройств
- `c` файлы символьных устройств
- `p` именованные каналы

Монтирование файловых систем

Включение файловой системы в общее дерево каталогов – корень монтируемой ФС отожествляется с папкой в общем дереве (точка монтирования).

Для монтирования ФС обычно нужны права суперпользователя, обойти это ограничение можно с помощью FUSE (Filesystem in USErspace).

Для (раз-)монтирования ФС используется утилита `mount`.

Filesystem Hierarchy Standard

- /bin – важные программы
- /boot – файлы загрузчика
- /dev – файлы устройств
- /etc – файлы конфигурации
- /lib – библиотеки
- /media – точки монтирования отключаемых носителей
- /mnt – временные точки монтирования
- /opt – внесистемные пакеты
- /run – данные о процессе выполнения программ
- /sbin – важные системные программы
- /srv – данные сервисов, предоставляемых системой
- /tmp – временные файлы
- /usr – второй уровень иерархии
- /var – разные данные
- /home – *домашние папки пользователей
- /root – *домашняя папка суперпользователя

Процессы

Что такое процесс

Процесс – абстракция выполняющейся программы. Включает в себя адресное пространство (страницы памяти) и структуры ядра ОС.

Поток – контекст выполнения процесса. Потоки процесса делят общее адресное пространство.

Каждому потоку присвоен уникальный идентификатор PID. Потоки одного процесса имеют общий TGID (Thread Group ID).

Пример многопоточного процесса

```
top - 19:06:47 up 415 days,  3:25,  1 user,  load average: 0.17, 0.12, 0.06
Threads: 772 total,   1 running, 771 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.5 us,  0.1 sy,  0.0 ni, 99.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  64298.2 total,   2354.4 free,   6073.9 used,  55869.9 buff/cache
MiB Swap:  30518.0 total,  30017.5 free,    500.5 used.  57535.6 avail Mem

    PID    PPID nTH    TGID USER      NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
3136433       1   1 3136433 opendkim   0   23116    960    648 S   0.0   0.0   0:00.00 /sbin/opendkim
3136434 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   2:20.41 /sbin/opendkim
3136435 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   0:00.00 /sbin/opendkim
3136436 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   0:00.00 /sbin/opendkim
3136437 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   2:02.18 /sbin/opendkim
3138090 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   1:31.11 /sbin/opendkim
3704485 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   3:12.84 /sbin/opendkim
 285358 3136433   7 3136434 opendkim   0  555896  22016   4744 S   0.0   0.0   0:23.19 /sbin/opendkim
```

Информация о процессе

- идентификаторы
- состояние
- приоритет
- аргументы командной строки и окружение
- аттрибуты контроля доступа
- открытые файловые дескрипторы
- статистика использования памяти
- ограничения на ресурсы
- исполняемый файл и рабочая директория
- стек вызовов
- ...

Файловая система procfs

```
# TIME_STYLE='+%R' ls -l /proc/22130
dr-xr-xr-x  2 root root 0 21:08 attr
-rw-r--r--  1 root root 0 21:08 autogroup
-r--------  1 root root 0 21:08 auxv
-r--r--r--  1 root root 0 21:08 cgroup
--w-------  1 root root 0 21:08 clear_refs
-r--r--r--  1 root root 0 21:08 cmdline
-rw-r--r--  1 root root 0 21:08 comm
-rw-r--r--  1 root root 0 21:08 coredump_filter
-r--r--r--  1 root root 0 21:08 cpuset
lrwxrwxrwx  1 root root 0 21:08 cwd -> /
-r--------  1 root root 0 21:08 environ
lrwxrwxrwx  1 root root 0 21:08 exe -> /bin/bash
dr-x------  2 root root 0 21:08 fd
dr-x------  2 root root 0 21:08 fdinfo
-rw-r--r--  1 root root 0 21:08 gid_map
-r--------  1 root root 0 21:08 io
-r--r--r--  1 root root 0 21:08 limits
-rw-r--r--  1 root root 0 21:08 loginuid
dr-x------  2 root root 0 21:08 map_files
-r--r--r--  1 root root 0 21:08 maps
-rw-------  1 root root 0 21:08 mem
-r--r--r--  1 root root 0 21:08 mountinfo
-r--r--r--  1 root root 0 21:08 mounts
-r--------  1 root root 0 21:08 mountstats
dr-xr-xr-x 59 root root 0 21:08 net
dr-x--x--x  2 root root 0 21:08 ns
-r--r--r--  1 root root 0 21:08 numa_maps
-rw-r--r--  1 root root 0 21:08 oom_adj
-r--r--r--  1 root root 0 21:08 oom_score
-rw-r--r--  1 root root 0 21:08 oom_score_adj
-r--------  1 root root 0 21:08 pagemap
-r--------  1 root root 0 21:08 patch_state
-r--------  1 root root 0 21:08 personality
-rw-r--r--  1 root root 0 21:08 projid_map
lrwxrwxrwx  1 root root 0 21:08 root -> /
-rw-r--r--  1 root root 0 21:08 sched
-r--r--r--  1 root root 0 21:08 schedstat
-r--r--r--  1 root root 0 21:08 sessionid
-rw-r--r--  1 root root 0 21:08 setgroups
-r--r--r--  1 root root 0 21:08 smaps
-r--r--r--  1 root root 0 21:08 smaps_rollup
-r--------  1 root root 0 21:08 stack
-r--r--r--  1 root root 0 21:08 stat
-r--r--r--  1 root root 0 21:08 statm
-r--r--r--  1 root root 0 21:08 status
-r--------  1 root root 0 21:08 syscall
dr-xr-xr-x  3 root root 0 21:08 task
-r--r--r--  1 root root 0 21:08 timers
-rw-rw-rw-  1 root root 0 21:08 timerslack_ns
-rw-r--r--  1 root root 0 21:08 uid_map
-r--r--r--  1 root root 0 21:08 wchan
```

Возникновение новых процессов

fork – устаревший системный вызов клонирования процесса

clone – современный системный вызов, позволяет тонко контролировать разделяемые ресурсы

execve – системный вызов, заменяющий выполняемую программу

В Linux не происходит реального клонирования адресного пространства процесса, копирование осуществляется при попытке изменения данных (copy-on-write).

Группы процессов, сессии

Используются для контроля выполнения задач в командной оболочке.

- Группа процессов – процессы с одинаковым PGID, выполняющие одну задачу shell, т.е. формирующие один конвейер.
- Процессы с одинаковым SID формируют пользовательскую сессию – одну или несколько задач, разделяющих управляющий терминал.
- Только одна задача из сессии может выполняться в активном режиме, остальные выполняются в фоновом.

Лидер группы и лидер сессии – процессы, PID которых совпадает с PGID и SID.

Сигналы

Сигналы – основной механизм управления процессами.

Процесс может послать другому процессу один из 64 сигналов (около половины имеют предопределенное значение).

Процесс может:

- игнорировать сигнал;
- зарегистрировать функцию-обработчик сигнала (будет вызываться при его получении);
- выполнить стандартное для сигнала действие.

Некоторые сигналы нельзя игнорировать или обрабатывать.

Для отправки сигнала есть системный вызов `kill` (и одноименная утилита).

Пользователи

Пользователи

У каждого пользователя в системе есть числовой идентификатор – **uid**.

- Каждый процесс запущен от имени какого-либо пользователя.
- У каждого объекта в файловых системах есть владелец.
- Эти атрибуты используются для контроля доступа.
- В базе пользователей хранится дополнительная информация.

Группы пользователей

Каждый пользователь принадлежит одной или нескольким группам с числовыми **gid**.

- Каждый процесс запущен от имени какой-либо группы пользователей.
- У каждого объекта в файловых системах есть группа-владелец.
- В системе есть база пользовательских групп, для каждой группы указан список пользователей.

Типы пользователей и групп

- uid 0 – root, суперпользователь
- uid 65534 – nobody, не должен иметь в системе никаких прав
- gid 65534 – nogroup, групповой аналог nobody
- uid < 1000 – системные пользователи
- gid < 1000 – системные группы
- остальные uid и gid можно присваивать обычным пользователям

Базы пользователей

Системные базы passwd и group могут храниться разными способами.

- одноименные файлы в /etc/
- каталог LDAP
- NIS
- домен Windows (с помощью winbind)
- реляционная база данных
- ...

За выбор источника отвечает NSS (Name Service Switch, /etc/nsswitch.conf).

Какие данные хранятся

Пример записей passwd:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
user:x:1000:1000:User,,,:/home/user:/bin/bash
_apt:x:104:65534::/nonexistent:/bin/false
sshd:x:117:65534::/run/sshd:/usr/sbin/nologin
```

Пример записей group:

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
cdrom:x:24:user
sudo:x:27:user
audio:x:29:pulse,user
nogroup:x:65534:
user:x:1000:
ntp:x:110:
scanner:x:118:saned,user
```

Получить информацию из системных баз можно с помощью утилиты `getent`.

Аутентификация и авторизация

authentication	аутентификация – проверка подлинности пользователя
authorization	авторизация – проверка прав доступа
accounting	учёт – журналирование событий AA и учет используемых ресурсов

В современных Linux для AAA используются PAM.

PAM

Pluggable Authentication Modules – набор разделяемых библиотек, предоставляющих высокоуровневый интерфейс AAA для приложений и сервисов.

Задачи AAA разделены на группы:

- account – авторизация,
- auth – аутентификация,
- session – подготовка и учет,
- password – обновление данных, используемых для аутентификации.

Пример конфигурации PAM

/etc/pam.d/login

```
auth       optional   pam_faildelay.so  delay=3000000
auth       requisite  pam_nologin.so
session    [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so close
session    required   pam_loginuid.so
session    optional   pam_motd.so motd=/run/motd.dynamic
session    optional   pam_motd.so noupdate
session    [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
session    required   pam_env.so readenv=1
session    required   pam_env.so readenv=1 envfile=/etc/default/locale
@include common-auth
auth       optional   pam_group.so
session    required   pam_limits.so
session    optional   pam_lastlog.so
session    optional   pam_mail.so standard
session    optional   pam_keyinit.so force revoke
@include common-account
@include common-session
@include common-password
```