Практическое занятие по курсу СПО

Знакомство с операционными системами семейства *nix на примере ОС ALT Linux Server

Рассматриваемые темы

- История развития ОС семейства Unix
- Общее описание устройства ЭВМ и центрального процессора
- Порядок начальной загрузки ЭВМ
- Дисковые устройства, разделы дисков, таблицы разделов
- Файловые системы, их типы и общее устройство
- Порядок загрузки операционной системы
- Архитектура операционных систем *nix
- Порядок запуска и выполнения процессов
- Пользователи и процессы
- Права доступа к файлам
- Организация файловой системы *nix
- Стандартная структура файловой системы
- Командный интерпретатор и его роль в системе
- Основные команды системы
- Работа с программными компонентами
- Менеджеры пакетов
- Системы инициализации и управление работой демонов

История развития ОС семейства Unix

- 1964 – AT&T, GE и MTI начали разработку MULTICS
- 1969 – AT&T (Bell Labs) выходит из проекта MULTICS
- 1969 – Ken Thompson, Dennis Ritchie, Douglas McIlroy, первая версия UNIX для PDP-7
- 1971, ноябрь – версия для PDP-11 (Edition 1)
- 1969-1973 – создание C
- 1973 – Edition 4, с ядром на C
- 1975 – Edition 5, полностью на C
- 1974 – распространение по университетам
- 1978 – BSD UNIX
- 1980 – начало коммерциализации UNIX, появление многочисленных ветвей системы
- 1988 – стандартизация систем, POSIX

Свободное программное обеспечение

1983 – Richard Stallman, манифест проекта GNU (GNU's Not Unix)

Свободы пользователей программ:
0. Свобода запускать программу в любых целях
1. Свобода изучения работы программы и адаптации её
2. Свобода распространять копии
3. Свобода улучшать программу и публиковать улучшения

Свободные лицензии:
- GNU General Public License, v. 3 (GNU GPL v.3)
- GNU General Public License, v. 2 (GNU GPL v.2)
- GNU Free Documentation License
- BSD
- MIT
- Artisitic (Perl license)
- MPL (Mozilla Public License)
- ...

Linux-системы

- 1987 – MINIX
- 1991, сентябрь – анонс разработки Linux
- 1991, 5 октября – 0.02
- 1994, 6 марта – 1.0

- Состав Linux-системы:
  ядро + утилиты GNU + сторонние приложения

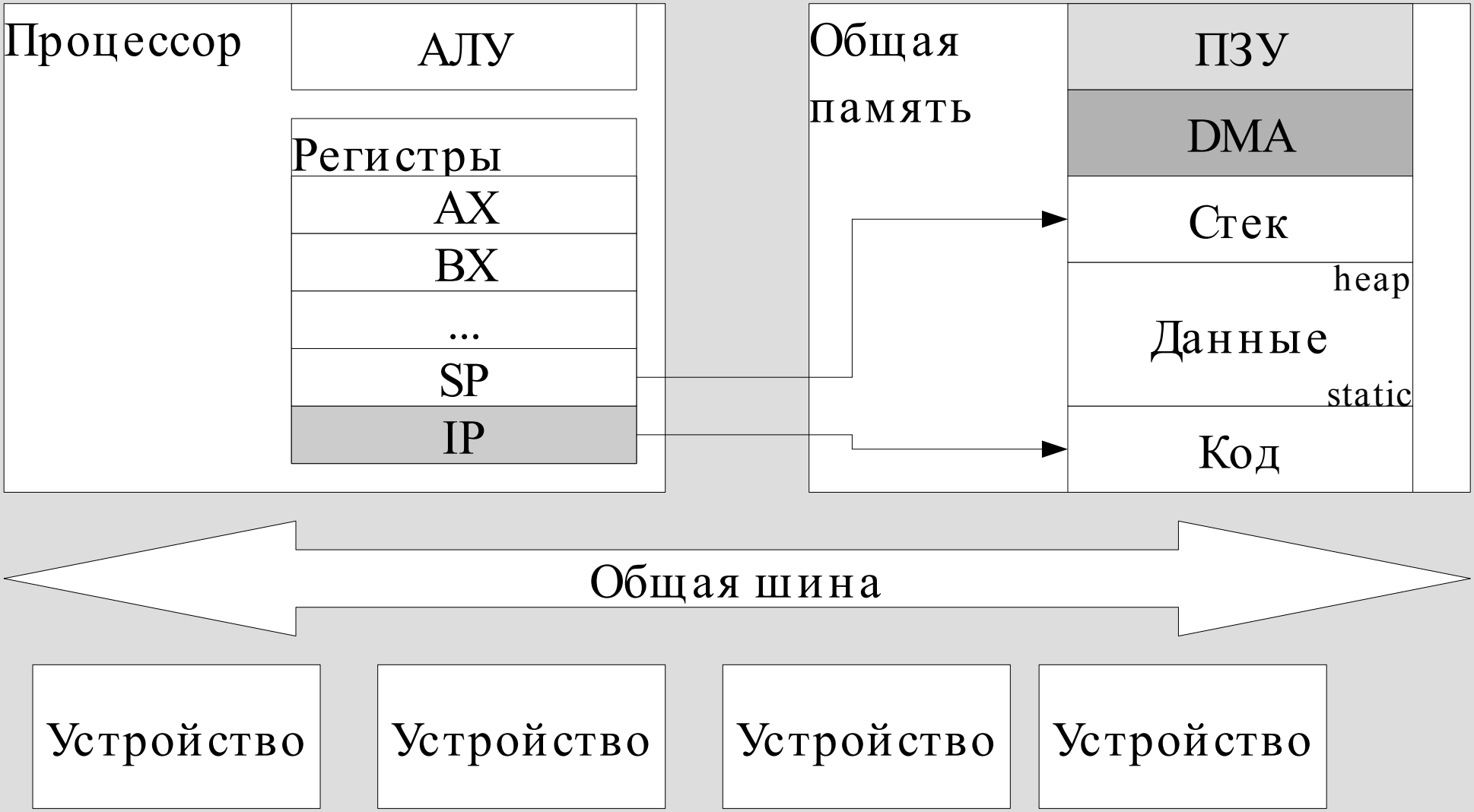
- Дистрибутивы:
  Debian, Ubuntu, RedHat, CentOS, Mandriva, Suse, Slackware,
  ALT Linux, ASTRA Linux, ...

Устройство ЭВМ

ЭВМ:

- Процессор
  - Арифметически-логическое устройство
  - Регистры
- ОЗУ
- Периферийные устройства
  - Устройства ввода-вывод
  - Накопители данных
  - Сетевые карты
  - Графические адаптеры
  - ….

Устройство ЭВМ



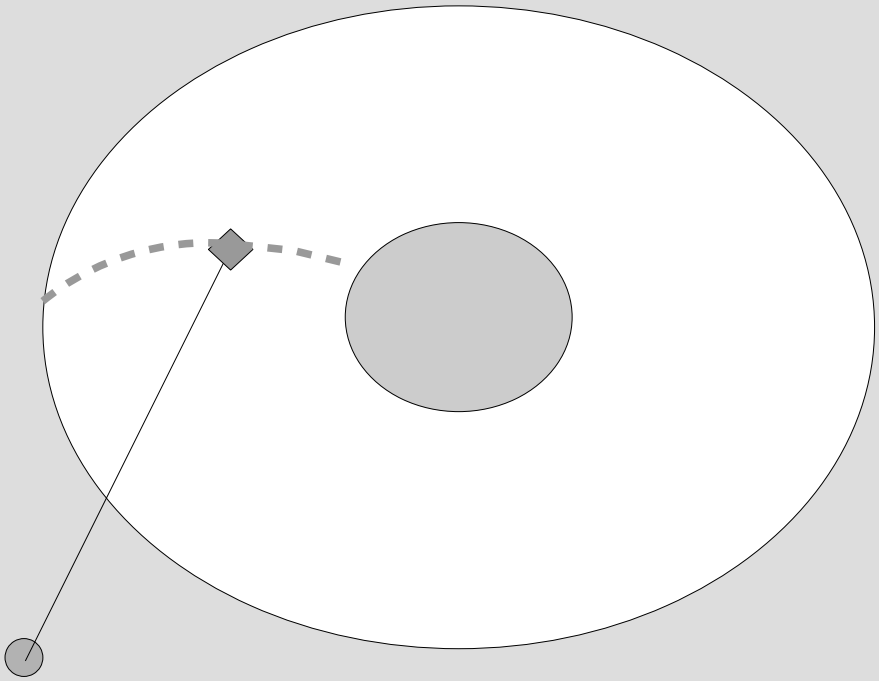
Загрузка ЭВМ

- Включение питания
- Инициализация процессора
- Начало выполнение кода из ПЗУ
  - Monitor
  - BIOS (Basic Input/Output System)
  - UEFI (Unified Extensible Firmware Interface)
- Инициализация ОЗУ
- Инициализация периферийных устройств
- Определение загрузочного устройства
- Выполнение первичного загрузчика

Дисковые устройства

Накопители данных:

- Блочные устройства
- HDD, SSD, оптические диски, сетевые диски, виртуальные диски, дисковые массивы, …



Геометрия дисков:

- CHS (Cylinder-head-sector)
  - S: 1..63
  - H: 0..255
  - C: 0..65535
- LBA (Logical Block Addressing)

Разделы дисков

Таблицы разделов: MBR, GPT.

- MBR (Master Boot Record) – 4 раздела
  (1 расширенный на 4 дополнительных раздела)
- GPT (GUID Partition Table) – 128 разделов

Дисковые массивы:

- RAID-0 (два диска вместе)
- RAID-1 (два диска с зеркальной записью)
- RAID-5 (N дисков с данными + 1 диск с контрольной суммой)
- RAID-6 (N дисков с данными + 2 диска с контрольной суммой)
- RAID-10 (RAID-0 из RAID-1), RAID-50, RAID-60, ...

Программные дисковые массивы

Менеджеры дисковых томов: LVM

Типы файловых систем

Общего назначения:

- Ext2
- Ext3
- Ext4
- Btrfs
- XFS
- JFS
- ReiserFS
- ISOFS (iso9660)
- UDF
- VFAT
- NTFS

Псевдо-файловые системы

- procfs
- sysfs
- udevfs

Сетевые файловые системы

- NFS
- CIFS

Специализированные файловые системы

- tmpfs
- jffs2
- squashfs

Устройство файловых систем

FAT:
- таблица размещения файлов
- корневой каталог
- подкаталоги

EXT2:
- inode:
  - список блоков с данными файла
  - метаинформация: владелец, группа владельца, права доступа, даты создания, изменения и последнего доступа, количество ссылок на файл, расширенные аттрибуты
- каталоги:
  - имя файла и его тип
  - номер inode для файлов и каталогов

Устройство файловых систем

Типы файлов:
- обычный файл (file)
- каталог (directory)
- символьная ссылка (soft link)
- устройство (device): символьное, блочное (char, block)
- именованные каналы ввода-вывода (named pipe)
- сокеты (socket)
- двери (door)

Жёсткие ссылки (hard link) – записи в каталогах для inode

Отказоустойчивость файловых систем:
- ➔ Журналируемые файловые системы: ext3, ext4, XFS, …
- ➔ Файловые системы CoW: BtrFS, XFS+расширение reflink

Фрагментация файловых систем
Снимки состояния файловых систем

Загрузка операционной системы

- Чтение и запуск первичного загрузчика с диска
  - Первый блок для MBR
  - Загрузочный раздел с FAT для UEFI
- Чтение и запуск вторичного загрузчика

- Определение раздела диска с ядром операционной системы
- Чтение ядра операционной системы
- Чтение initrd (initial ramdisk)

- Инициализация ядра операционной системы
- Загрузка модулей (драйверов) основных устройств
- Определение корневой файловой системы
- Замена initrd на корневую файловую систему

- Запуск первого процесса (/sbin/init, PID=1, UID=0, GID=0).
- Запуск системы инициализации операционой системы

Операционные системы

Операционные системы:

- Однозадачные
- Многозадачные
  - Кооперативная многозадачность
  - Вытесняющая многозадачность

Операционные системы:

- Однопользовательские
- Многопользовательские
  - 1 пользовательский сеанс + фоновые задачи
  - N пользовательских сеансов + фоновые задачи

Unix – многозадачная многопользовательская ОС

Архитектура Unix-систем

GUI *KDE Gnome ...* *X.org*	Пользовательские программы *vim, mc, ...*	Демоны *Apache, Postfix* *PostgreSQL, MySQL ...*
Компилятор C	Базовые утилиты	Оболочки
Стандартная библиотека C		Прочие библиотеки

Файлы устройств	**Ядро**	Сокеты
Процессы	Файловые системы	Протоколы
Управление памятью		Драйверы устройств

Оборудование

Общий вид запуска программ

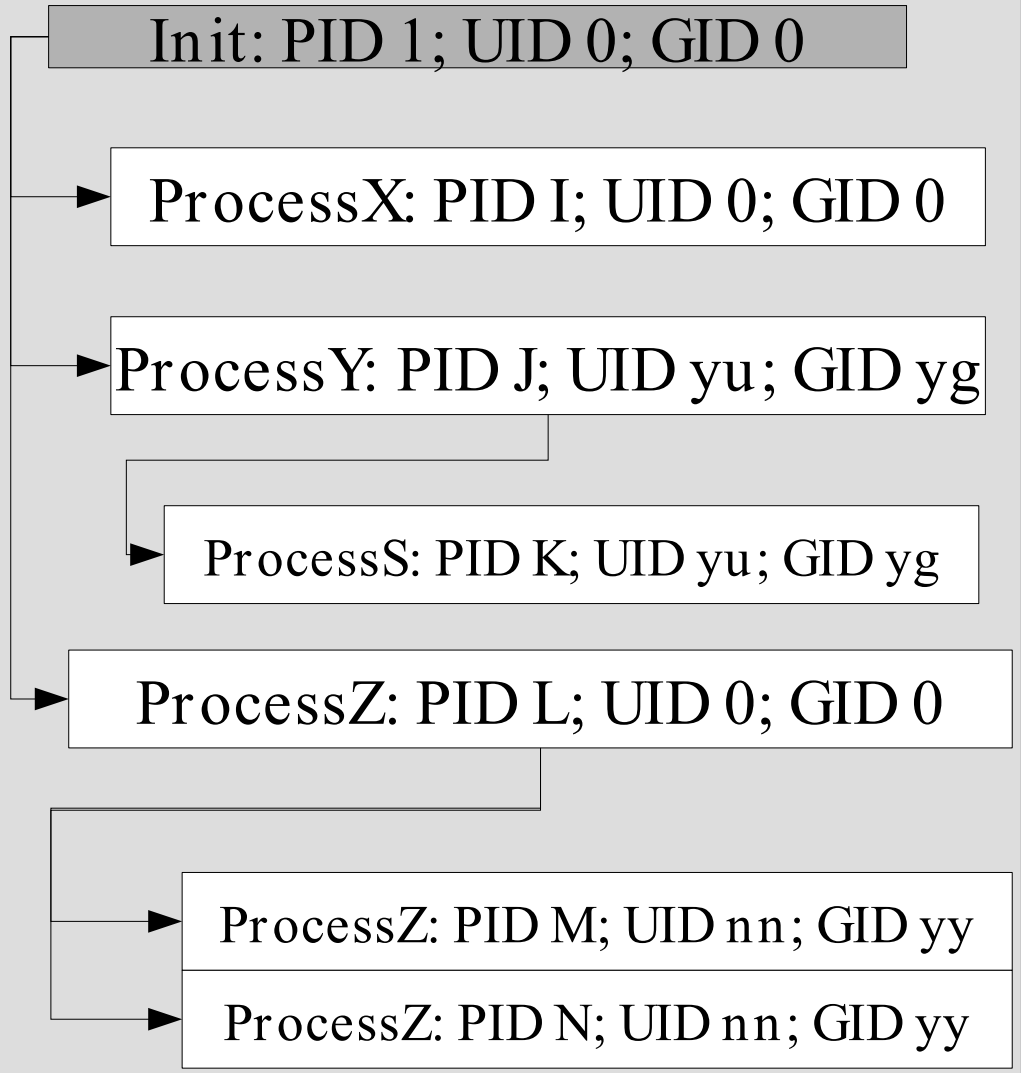
```
int main(void)
{
 pid_t pid = fork();

 if (pid == -1){ perror("fork failed"); exit(1);}
 else
   if (pid == 0) {
      printf("Hello from the child process!\n");
      exit(0);
   } else {
      int status;
      (void)waitpid(pid, &status, 0);
   }
 return 0;
}
```

Общий вид запуска программ

Многозадачная система:
- параллельно выполняется много программ
- ядро ведёт список процессов

Программы запускают другие программы:
- процессы-родители
- дочерние процессы

Многопользовательская система:
- каждый процесс принадлежит определённому пользователю
- ядро учитывает пользователей при выполнении программой определённых действий



Пользователи и процессы в системе

- Каждый процесс выполняется с правами определённого пользователя
- Система различает пользователей и группы пользователей
- UID, GID – числовые идентификаторы пользователя и группы
- /etc/passwd – список пользователей в системе
- /etc/group – список групп в системе

- Пользователи делятся на обычных и псевдо-пользователей
- Пользователь с UID=0 – администратор системы
- Традиционное имя для пользователя с UID=0 - root

- Первый процесс в системе – init. Запускается ядром с UID=0, GID=0
- Есть системные вызовы для смены UID и GID
- Изменить свой UID может только процесс с UID=0

Права доступа к файлам

Каждый файл имеет владельца, группу, и права доступа
Права доступа:
•Read – право на чтение из файла
•Write – право на запись в файл
•eXecute – право на выполнение файла

Каталоги также имеют владельца, группу, и права доступа
•Read – право на чтение списка файлов в каталоге
•Write – право на запись в каталог (создание/удаление файлов)
•eXecute – право на переход в каталог

Запись прав:
• rwx – право на чтение, запись и выполнение
• rw- – право на чтение и запись
• r-- – право на чтение и запись

Права доступа к файлам

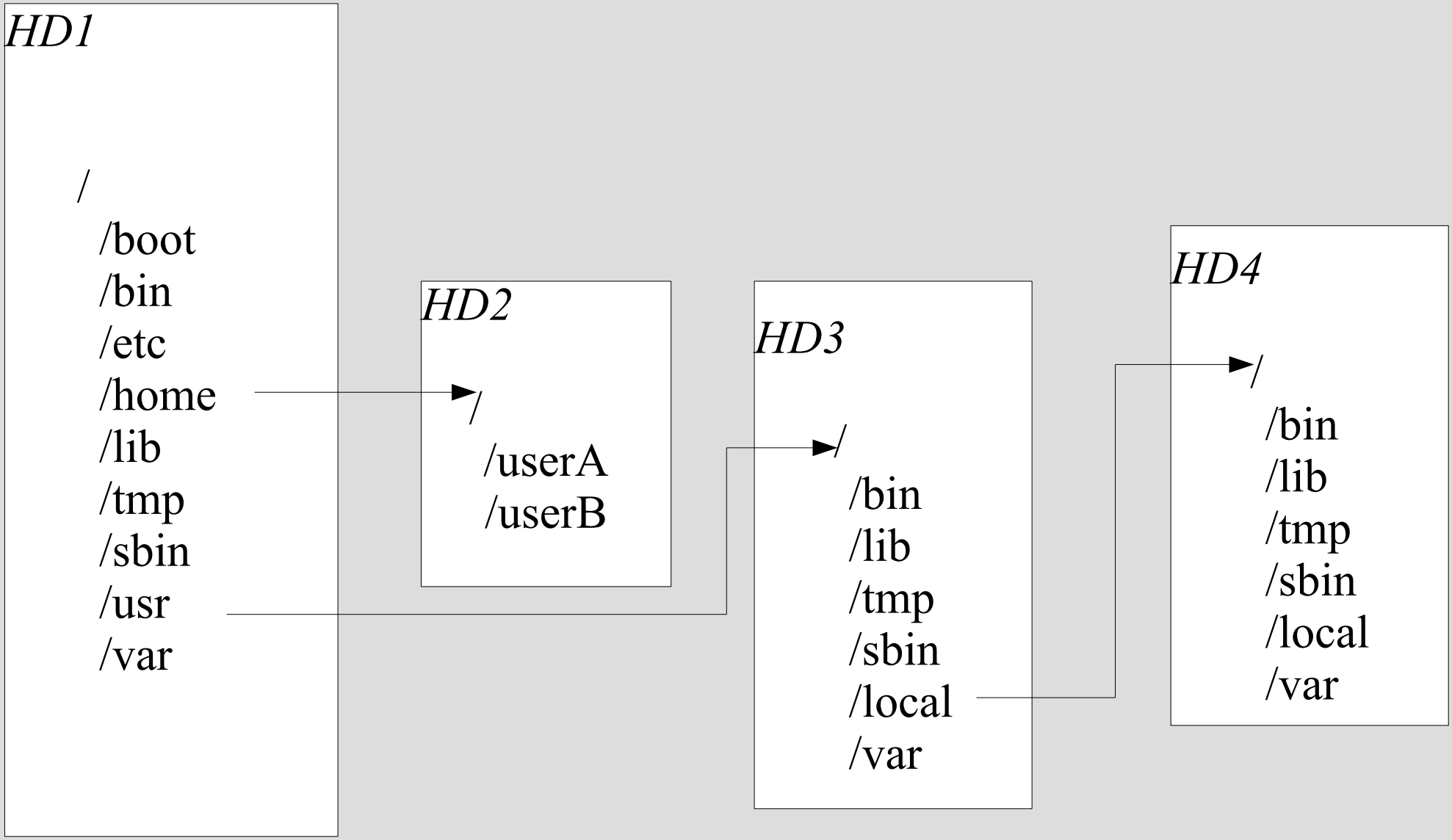
```
$ ls -l ~
итого 8
drwx------ 2 student student 4096 Фев 19 17:30 Documents
-rw-r--r-- 1 student student    0 Фев 20 08:03 file.txt
drwx------ 2 student student 4096 Фев 19 15:59 tmp
$ ls -l /var
итого 72
drwxr-xr-x  2 root root   4096 Апр 19  2007 adm
drwxr-xr-x  4 root root   4096 Фев 15 08:32 cache
drwxr-xr-x  2 root root   4096 Апр 19  2007 db
dr-xr-xr-x  2 root root   4096 Апр 19  2007 empty
drwxr-xr-x 11 root root   4096 Фев  9 15:29 lib
drwxr-xr-x 14 root root   4096 Фев 20 07:32 log
lrwxrwxrwx  1 root root     10 Фев  5 13:22 mail -> spool/mail
drwxr-x---  2 root nobody 4096 Апр 19  2007 nobody
drwxrwxrwt  2 root root   4096 Апр 19  2007 tmp
drwxr-xr-x  3 root root   4096 Фев 15 09:24 www
drwx------  2 root root   4096 Апр 19  2007 yp
$ ls -l /bin/su
-rws--x--- 1 root wheel 23712 Окт 18  2006 /bin/su
```

Права доступа к файлам

Биты доступа:
- SUID (Set User-ID)
- SGID (Set Group-ID)
- Sticky bit

```
$ ls -l /bin/su
-rws--x--- 1 root wheel 23712 Окт 18  2006 /bin/su

$ ls -l /usr/bin/crontab
-rwx--s--x 1 root crontab 39424 июн 30  2022 /usr/bin/crontab

$ ls -ld /var/spool/cron/
drwx-ws--T 2 root crontab 4096 фев 14 12:42 /var/spool/cron/
# ls -l /var/spool/cron/
total 4
-rw------- 1 webadmin crontab 1428 Feb 25 20:58 webadmin

$ ls -ld /tmp
drwxrwxrwt 21 root root 420 мар 16 01:11 /tmp
```

Стандартная структура файловой системы



Стандартная структура файловой системы

- структура файловой системы стандартизирована
- есть стандарные каталоги для размещения тех или иных файлов

```
$ ls -1 /
bin
boot
dev
etc
home
lib
lib64
lost+found
media
mnt
opt
...
....
proc
root
sbin
srv
sys
tmp
usr
var
```

```
$ ls -1 /usr/
bin
etc
games
include
lib
lib64
libexec
local
sbin
share
X11R6
```

```
$ ls -1 /var
...
empty
lib
local
lock
Log
...
run
spool
tmp
...
```

Командный интерпретатор и его роль в системе

- Обеспечивает пользовательский интерфейс командной строки
- Позволяет пользователю запускать программы
- Предоставляет возможность создания и исполнения файлов с последовательностями команд – скриптов
- Имеет набор встроенных команд
- Часть системных утилит, в т.ч. управления процессами запуска / остановки системы, написаны на языке командного интерпретатора

Существует ряд командных интерпретаторов:
sh, csh, tsh, bash, zsh ...

Общий формат вызова команды выглядит следующим образом:

```
$ command -f --flag --key=parameter argument1 agrument2 ...
```

Основные команды системы

Смена каталога:	cd
Просмотр каталога:	ls
Создание каталога:	mkdir
Удаление каталога:	rmdir
Удаление файла:	rm
Изменение даты/создание пустого файла:	touch
Просмотр текстового файла:	cat
Поэкранный просмотр файла:	less
Просмотр начала файла:	head
Просмотр конца файла:	tail
Изменение прав доступа к файлу:	chmod
Изменение владельца файла:	chown
Редактор:	vi / vim

Выполнение программ

Получить список процессов:
$ ps; ps aux

Послать сигнал процессу:
$ kill -<signal> <pid>

Остановить процесс:
$ kill -SIGSTOP <pid>

Продолжить выполнение:
$ kill -SIGCONT <pid>

Список сигналов:
$ kill -l

SIGKILL –	уничтожить процесс
SIGTERM –	завершить процесс
SIGQUIT -	завершить процесс
SIGSTOP –	остановить процесс
SIGCONT -	продолжить процесс

Работа с программными компонентами

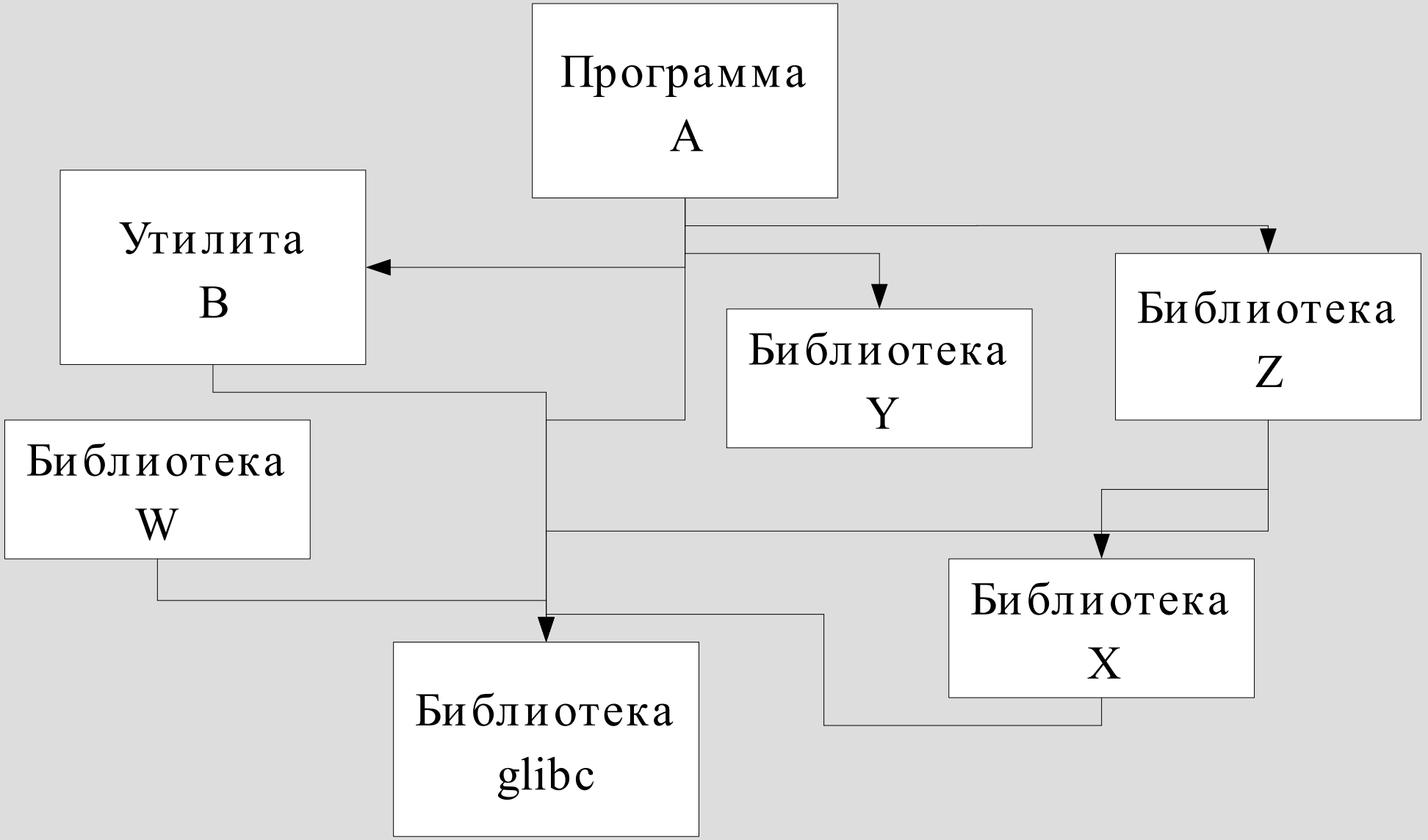
Основная часть программного обеспечения доступна в виде исходных кодов.

Дистрибутив – набор программных пакетов, настроенных для совместной работы в системе.

Существует большое число различных дистрибутивов:
- общего назначения: Debian, Ubuntu, RedHat, CentOS, Arch, Mandriva, ALT Linux, ASP Linux, ASTRA Linux.
- специального назначения: Mobian, Manjaro, OpenWRT, RIP, DBAN, …

Как правило, программные пакеты используют другие программные пакеты – в виде разделяемых библиотек, внешних исполняемых файлов, и т.п.

Работа с программными компонентами



Менеджеры пакетов

Менеджеры пакетов:
- .deb – dpkg (Debian Package manager) - Debian, Ubuntu
- .rpm – RPM Package Manager – RedHat, Fedora, Suse, ALT Linux, …

Основная утилита RPM – rpm

RPM отслеживает зависимости в отдельных пакетах .rpm.
Совокупность пакетов – репозиторий.
В репозитории (в идеале) зависимости между пакетами замкнуты.

В ALT Linux для организации репозитория используется
APT (Advanced Packaging Tool)

Менеджеры пакетов

Настройки APT: /etc/apt/sources.list, /etc/apt/sources.list.d/*

Запись о репозитории:

#rpm [cert] ftp://ftp.altlinux.org/pub/distributions/ALTLinux/c10f2/branch x86_64 classic
#rpm [cert] ftp://ftp.altlinux.org/pub/distributions/ALTLinux/c10f2/branch noarch classic

Работа с APT:

apt-get update
apt-get dist-upgrade
apt-get install <package>
apt-get remove <package>
apt-cache search <название>

Системы инициализации: sysvinit

Управлением порядком загрузки занимаются системы инициализации: systemd, sysvinit, Upstart, Runit, Launcd, Initng, ...

Уровни загрузки системы для sysvinit:
0 — уровень остановки системы
1 — однопользовательская система
2 — многопользовательская система без сетевой поддержки
3 — многопользовательская система
4 — предоставлено для конкретных систем
5 — многопользовательская система с поддержкой графики
6 — уровень перезагрузки системы
Переход между уровнями осуществляет командой init

Демон – традиционное название для неинтерактивных программ.
Запуск

Система инициализации sysvinit

sysvinit:
- действия при смене уровней загрузки выполняются скриптами командного интерпретатора
- размещение скиптов – в /etc/rc.d/
- запуск демонов – скриптами из /etc/rc.d/init.d/ (/etc/init.d),
- порядок запуска/остановки демонов для соответствующего уровня загрузки – символьные ссылки на скрипты демонов в /etc/rc.d/rc<N>.d/ ,
- запуск и остановка демонов выполняется последовательно.

Включение / выключение автоматической загрузки:
chkconfig <daemon> on; chkconfig <daemon> off

Добавление / удаление ссылок в /etc/rc.d/rc<N>.d/:
chkconfig <daemon> --add; chkconfig <daemon> --del

Запуск / остановка / состояние демона в ручном режиме:
service <daemon> start; service <daemon> stop
service <daemon> status

Система инициализации systemd

systemd:
- действия выполняются отдельным демоном systemd;
- описание действий – файлы конфигурации;
- порядок запуска/остановки указывается относительно других сервисов;
- запуск и остановка демонов выполняется по-возможности параллельно;

Включение / выключение автоматической загрузки:

```
# systemctl enable  [--now] <daemon>
# systemctl disable [--now] <daemon>
```

Запуск / остановка / состояние демона в ручном режиме:

```
# systemctl start  <daemon>; systemctl stop <daemon>
# systemctl status <daemon>
```

Помимо собственно системы инициализации в состав systemd входят: journald, udevd, logind, networkd, containerd, ...

Системные журналы

sysvinit: syslogd / syslogd-ng / rsyslogd
- отдельный независимый от sysvinit демон, получающий от программ и записывающий в файлы журналов записи;
- файлы журналов – текстовые файлы;
- просмотр журналов – стандартными утилитами;
- для ротации журналов требуется отдельный сервис logrotate, запускающийся по расписанию.

systemd: journald.
- входит в состав systemd;
- файлы журналов – бинарные, сжаты, с расширенной информацией о записях;
- просмотр журналов – отдельной утилитой journalctl;
- ротация журналов ведётся journald.

Демоны и приложения: могут использовать syslogd / journald, и/или вести журналы самостоятельно.