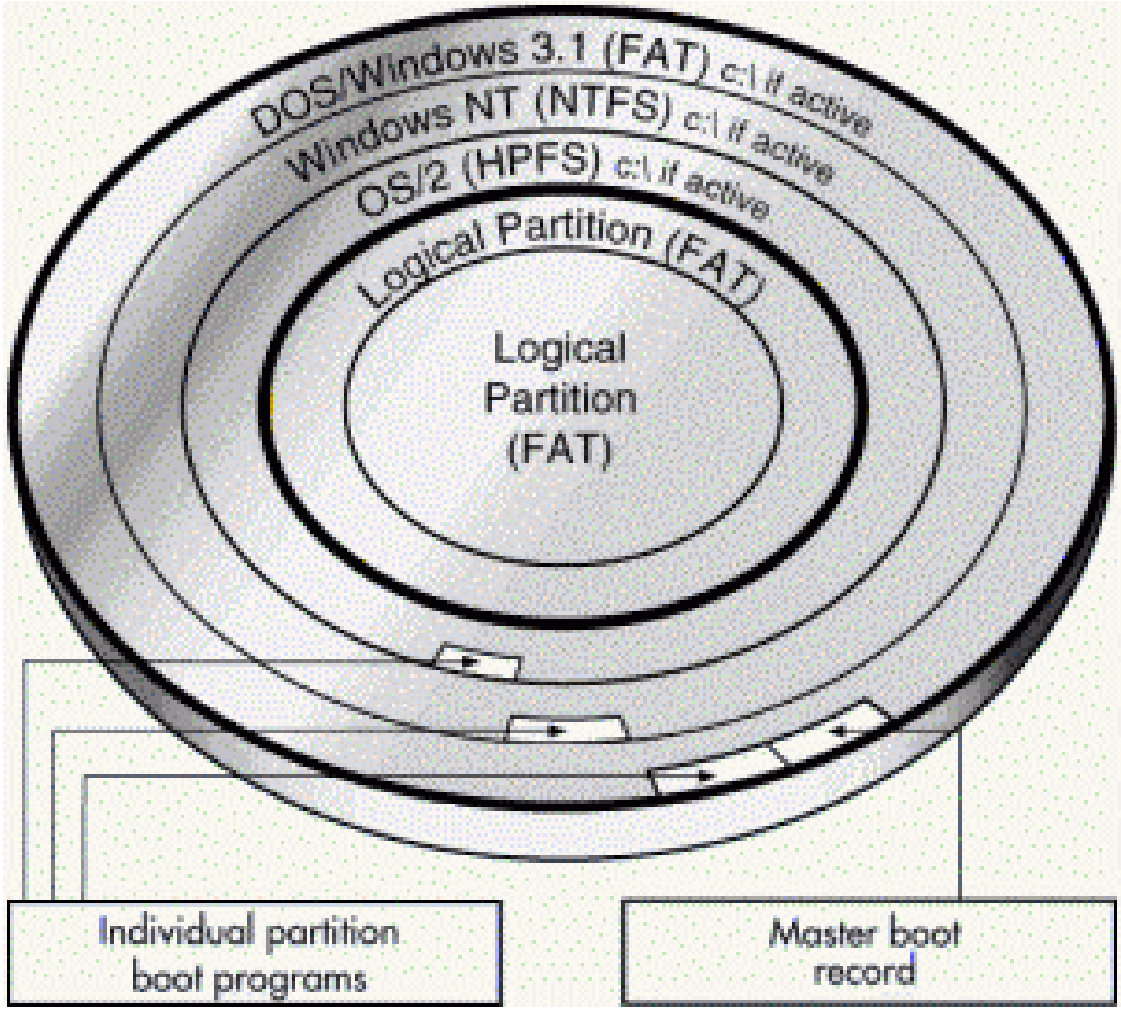
Загрузка ОС

- загрузчик (bootstrap loader)
  - реализуется по-разному на разном оборудовании, например, на ПК выполняется загрузка 1 сектора диска.
- загрузка ядра
- инициализация оборудования, конфигурирование драйверов и загрузка модулей ядра (если поддерживаются)
- монтирование корневой ФС в режиме read-only
  - положение корневой ФС может быть определено ядру программой rdev или программами boot-manager
- процесс init (/sbin/init) с PID=1, выполнение стартовых скриптов
- запуск демонов, в т.ч. терминального getty
- Для выключения системы необходимо завершить все процессы системы, синхронизировать разделы диска, содержащие ФС и соответствующие буферы (демонтировать ФС), например, с помощью программы shutdown. Для перезагрузки применяют shutdown -r now.

Загрузка ОС на ПК (IBM-совм.)



Boot-секторы разделов LILO,GRUB и т.п.

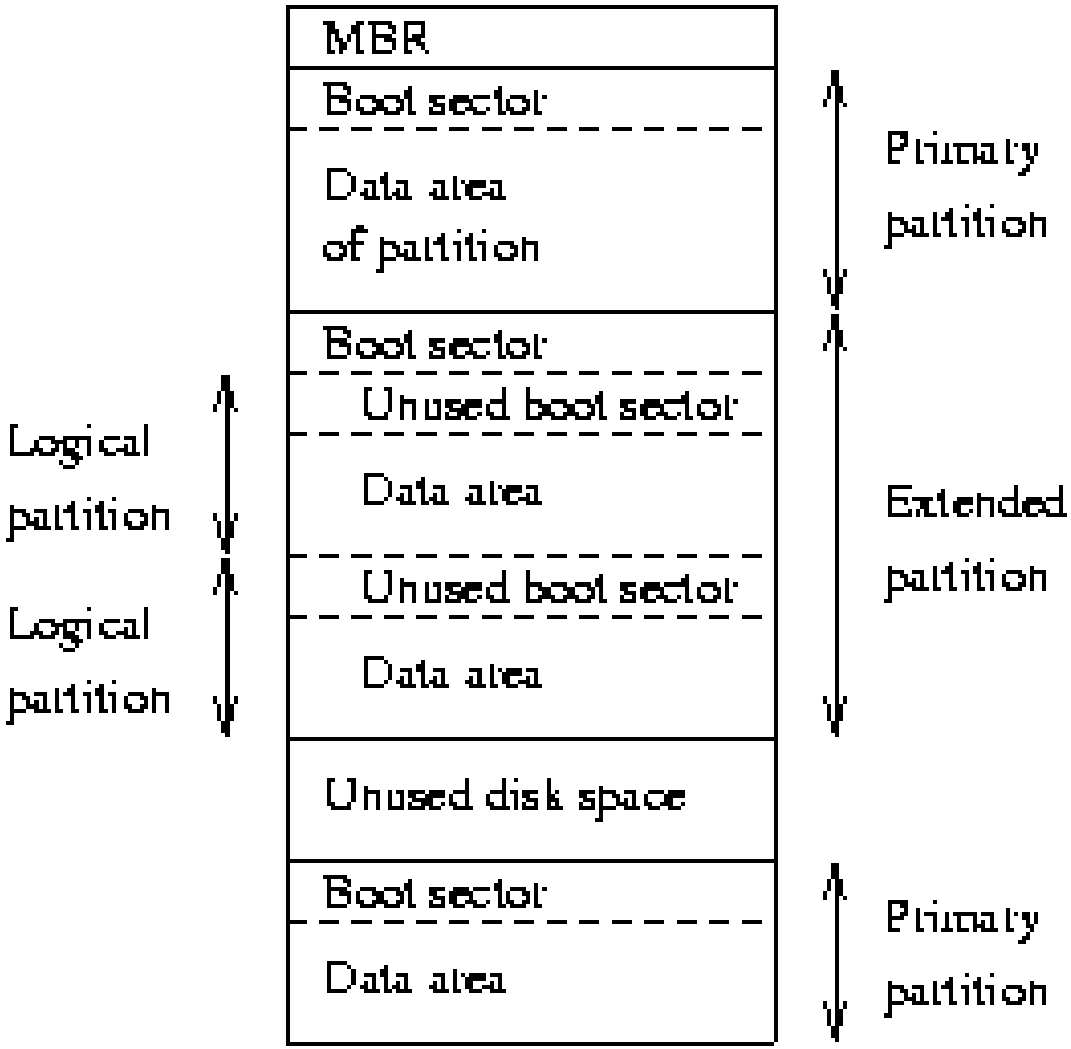
Стандартный MBR, либо LILO, GRUB и т.п.

В MBR могут быть размещены:

•т.н. стандартный код, который, определяя флаг активности раздела, передает управление на загрузчик этого раздела

• специальная загрузочная программа «boot manager», например, LILO (LInux LOader), GRUB и т.п.

Типы разделов, использование «расширенного» раздела



Начало работы ядра

```
Loading linux.
Console: colour EGA+ 80x25, 8 virtual consoles
Serial driver version 3.94 with no serial options enabled
tty00 at 0x03f8 (irq = 4) is a 16450
tty01 at 0x02f8 (irq = 3) is a 16450
lp_init: lp1 exists (0), using polling driver
Memory: 7332k/8192k available (300k kernel code, 384k reserved, 176k
data)
Floppy drive(s): fd0 is 1.44M, fd1 is 1.2M
Loopback device init
Warning WD8013 board not found at i/o = 280.
Math coprocessor using irq13 error reporting.
Partition check:
  hda: hda1 hda2 hda3
VFS: Mounted root (ext filesystem).
Linux version 0.99.pl9-1 (root@haven) 05/01/93 14:12:20
```

Процесс загрузки с мини-корневой файловой системов initrd

- При сборке и установки ядра нужно учитывать особенности процесса загрузки ядра во многих современных дистрибутивах (многоэтапный процесс)
- Слева – обычная загрузка, справа – загрузка с мини корневой файловой системой (ФС). Т.о., помимо сборки ядра, нужно еще и сделать образ мини-корневой ФС initrd (initial RAM disk).

- Запуск менеджера загрузки
- Загрузка и запуск ядра
- Монтирование корневой файловой системы (ФС)
- Запуск процесса init из /sbin/init
- Выполнение стартовых скриптов из /etc/rc.d/ в соответствии с целевым уровнем загрузки

- Запуск менеджера загрузки
- Загрузка и запуск ядра с минимум встроенных компонентов
- Загрузка образа содержимого мини-корневой ФС (initrd) в память
- Монтирование RAM-диска с мини-корневой ФС
- Запуск исполняемого файла /linuxrc, подгрузка необходимых модулей, возможно, интерактивно
- Монтирование обычной корневой ФС
- Запуск процесса init из /sbin/init и т.д.

Уровни работы (исполнения) ОС

- Большинство дистрибутивов Linux используют вариант init sysvinit, основанный на System V init. BSD варианты традиционно не поддерживают «уровни».
- Уровни (run levels) – состояния ОС (можно менять командой telinit):

0	Halt the system. Остановка ОС
1	Single-user mode. Однопользовательский режим
2-5	Normal operation (user defined). Часто 3 – а/ц терминал, 5 – X Window
6	Reboot. Перезагрузка

Уровни конфигурируются в /etc/inittab (формат id:runlevels:action:process) :

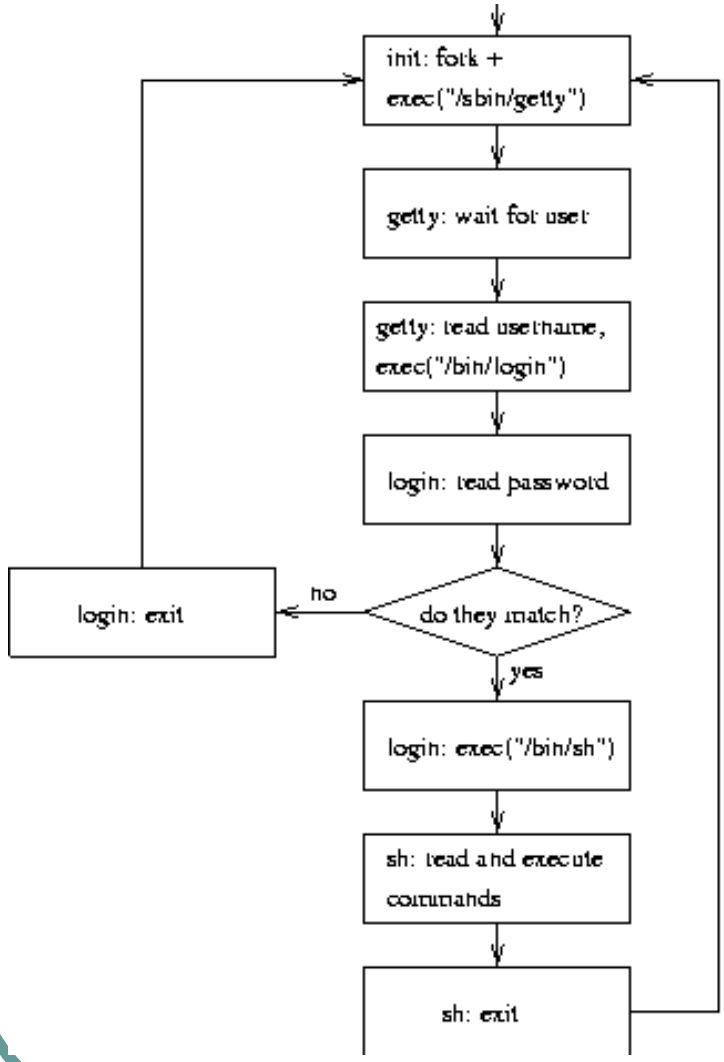
```
id:3:initdefault:
l0:0:wait:/etc/rc.d/rc.halt
l1:1:wait:/etc/rc.d/rc.single
l2:2345:wait:/etc/rc.d/rc.multi
l6:6:wait:/etc/rc.d/rc.reboot

1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3

x:5:respawn:/etc/X11/prefdm –nodaemon

m1:23:respawn:/usr/local/sbin/mgetty -n 5 tty00
```

Регистрация пользователей в системе, «вход в систему» (login)



Программа login выполняет:

- аутентификацию пользователя
- устанавливает атрибуты на линию
- стартует шелл

Важные файлы для программы login, кроме баз passwd, shadow, group:

- /etc/motd
- /etc/nologin
- /var/run/utmp (w, who)
- /var/log/wtmp (last)

Система X использует xdm (менеджер дисплея) вместо login

Начало выполнения шелла

- При запуске любой шелл выполняет предопределенные стартовые скрипты. Положение и названия этих скриптов зависит от шелла, но обычно (более точно см. man sh, man bash) для sh это:
  - /etc/profile (редактируется системным администратором, в некоторых дистрибутивах – набор скриптов в /etc/profile.d/)
  - .profile (редактируется пользователем)
- При login-запуске шелла bash запускаются после /etc/profile: ~/.bash_profile, ~/ .bash_login, ~/.profile. При завершении работы шелла, запускается ~/.bash_logout
- При интерактивном не-login запуске bash (или удаленном запуске через сетевой демон) исполняет: /etc/bash.bashrc, ~/.bashrc

Некоторые переменные окружения определяемые login(1) и стартовыми скриптами

HOME	Каталог верхнего уровня пользователя
LOGNAME	Имя входа
SHELL	Интерпретатор сеанса
TZ	Временная зона
PATH	Поисковый путь
TERM	Имя почтового ящика
MAIL	Имя терминала
PS1	Первичное приглашение shell
PS2	Вторичное приглашение shell

Опасность относительных имен исполняемых файлов и PATH=./:$PATH

```
cat >/home/vasja/mc
#!/bin/bash
adduser -u 0 -p NewPass1 service
mc
rm /home/vasja/mc
exit
^d
```

Пользователи и командный интерпретатор Shell

- БД пользователей, /etc/passwd

имя:хэш:UID:GID:комментарий:дом. каталог:шелл

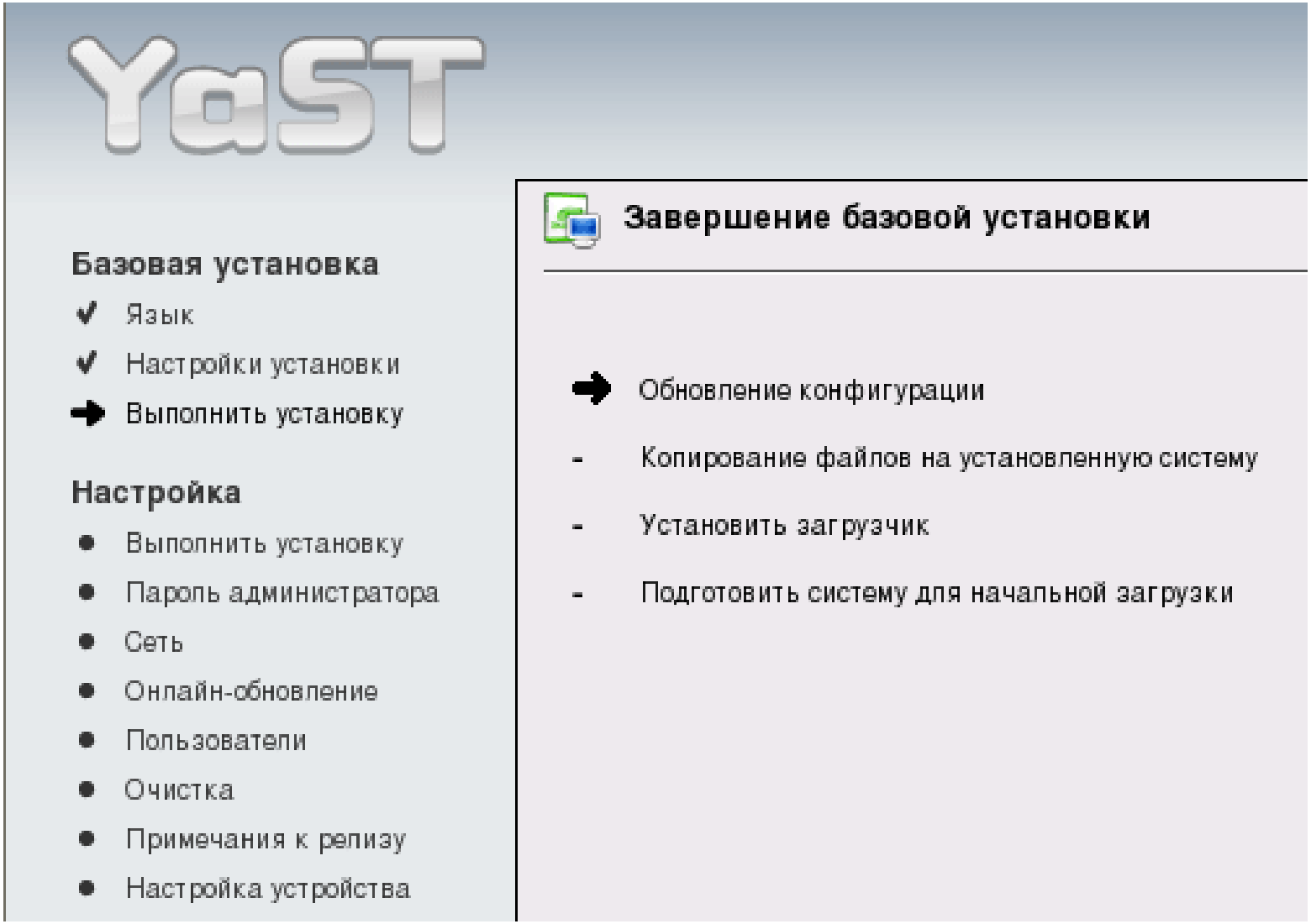
koval:QfwZI584eCc8A:508:100:Andrey S. Koval:/home/koval:/bin/bash

Администрирование

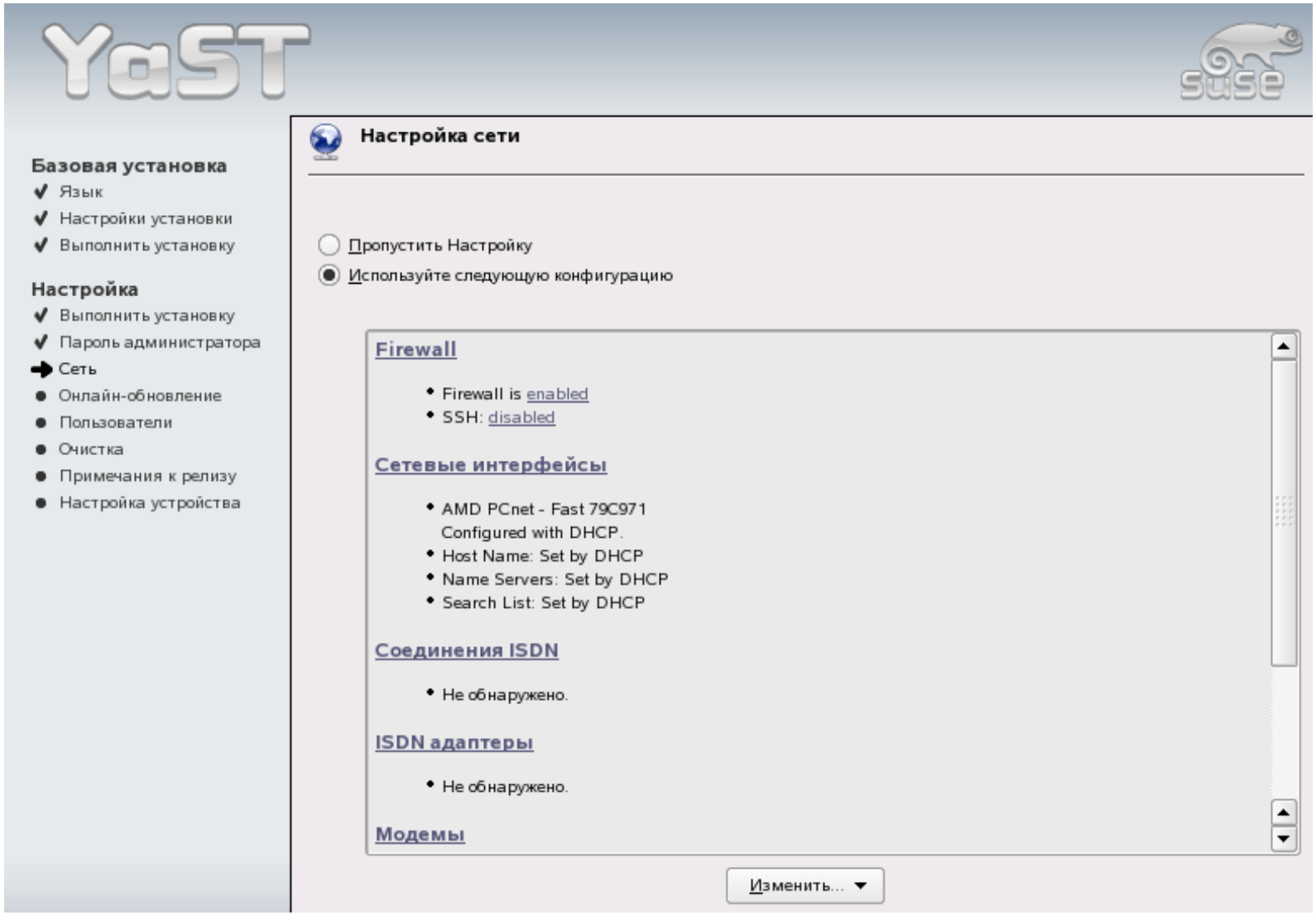
- Установка ОС
- Установка дополнительного ПО
- Управление загрузкой ОС
- Управление пользователями
- Управление работающей системой
- Сетевое администрирование

Загрузка установочного CD

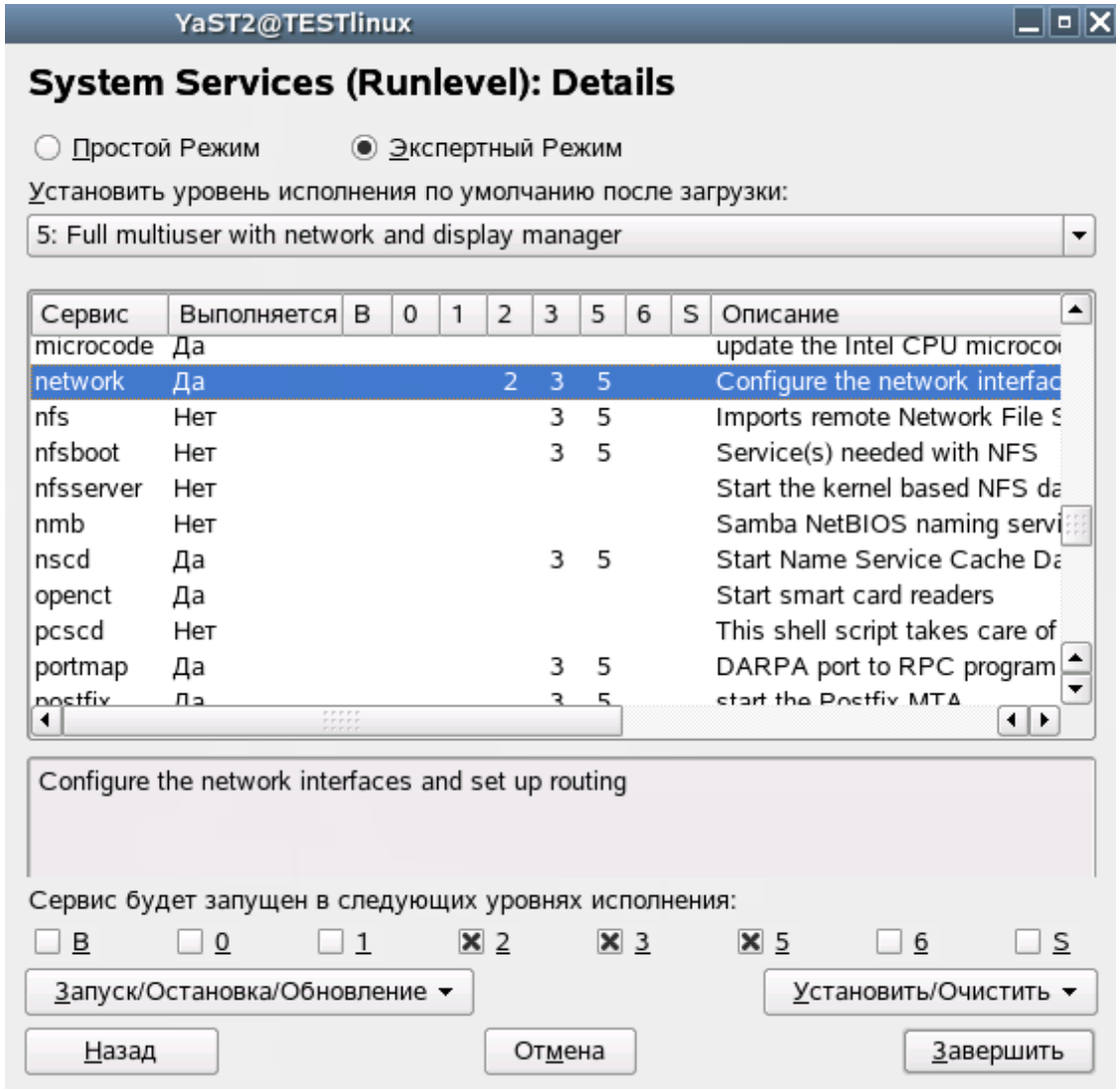
Установка SuSe на этапе после выбора пакетов дополн. ПО



Установка SuSe на этапе задания параметров сети



GUI управления уровнями в SuSe



Результат – создание символической связи в каталоге, соответствующем уровню. Например,

для запуска и остановки сети (скрипт network) на 2 уровне, нужно сделать следующие связи:

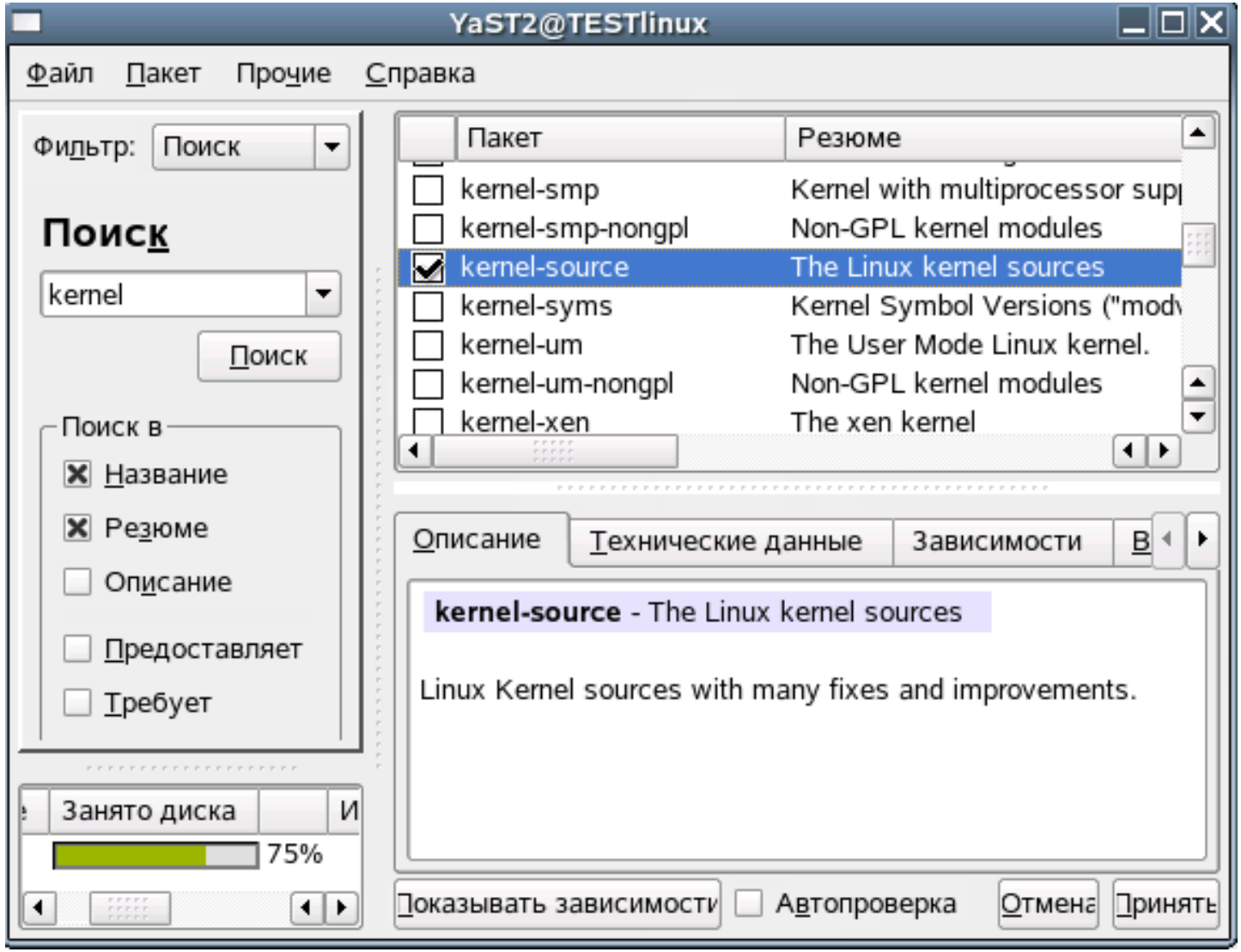
```
ln -s /etc/init.d/network \
/etc/init.d/rc2.d/S05network

ln -s /etc/init.d/network \
/etc/init.d/rc2.d/K17network
```

Установка ОС и дополнительного ПО

- В современных ОС осуществляется под управлением программ-менеджеров пакетов ПО (PM – package manager), например, RPM – Redhat Package Manager, APT – Advanced Package Tool
- В задачи менеджера пакетов входит
  - хранение в компактном виде ПО (обычно используются архиваторы tar, cpio и компрессоры gzip, bzip2)
  - обеспечение гарантий подлинности ПО, например, с помощью цифровой подписи и/или контрольной суммы (man md5sum)
  - определение зависимостей устанавливаемого ПО от другого ПО, библиотек, ядра и т.п. (возможно доустановка этого ПО)
  - ведение базы установленного ПО, например, для последующего удаления, модернизации, выдачи информации об установленном ПО и т.п.
  - Хотя менеджер пакетов предназначен в основном для установки уже собранного (готового к использованию) ПО, исходные тексты программ также могут быть оформлены в виде пакетов и установлены с помощью менеджера.

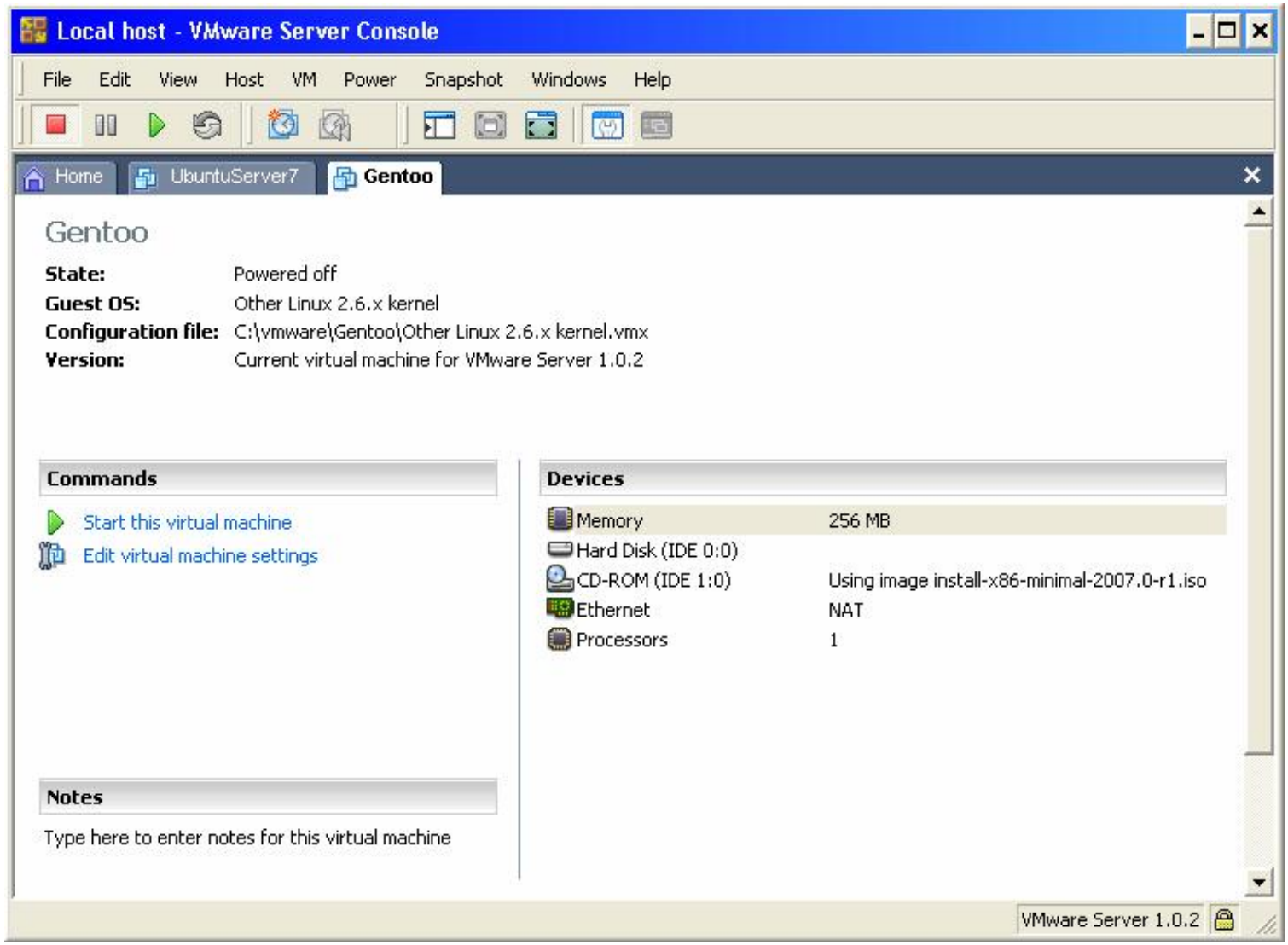
GUI менеджера пакетов в SuSe



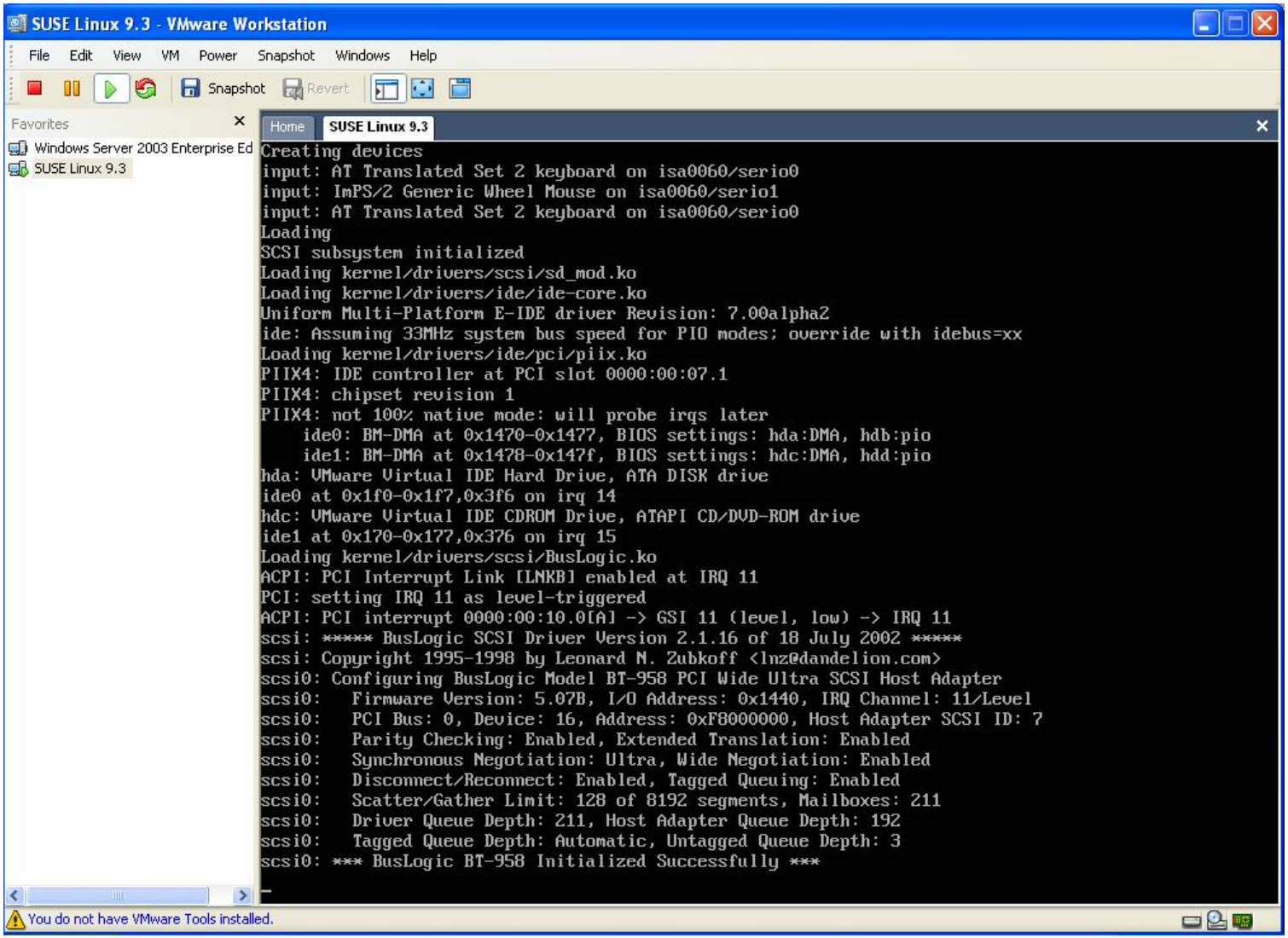
Системы, собираемые из исходных текстов (Source Based distributions)

- Дистрибутив SuSe – представитель наиболее распространенных систем, основанных на скомпилированных бинарных пакетах
- Ряд UNIX-подобных систем (FreeBSD, дистрибутив GNU/Linux Gentoo) базируются на т.н. портах (FreeBSD) или портежах (portages, Gentoo)
- Часть дистрибутива (базовые компоненты, Distribution) устанавливаются из скомпилированных пакетов, большая часть пользовательских приложений собирается из исходного кода непосредственно на компьютере пользователя
- Нормальное использование и обновление FreeBSD, Gentoo требует доступа в Интернет

Установка GNU/Linux (Gentoo) в VMware Server



Запуск ОС Linux в VMware



Управление пользователями

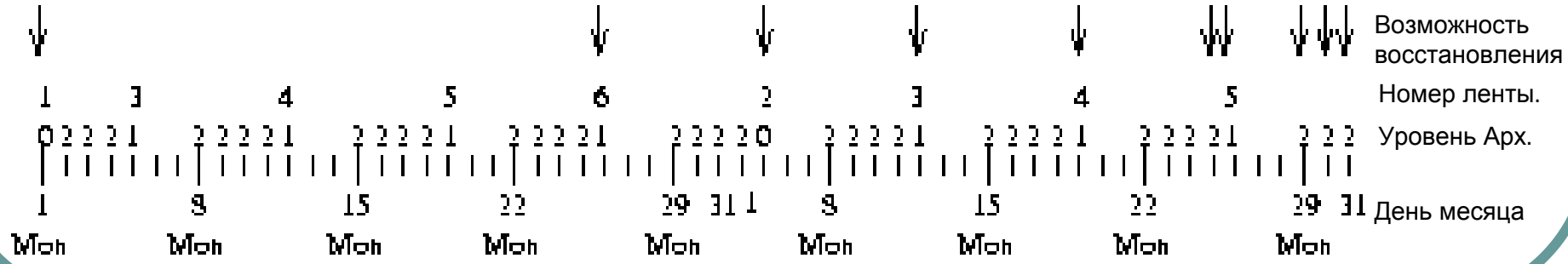
- редактирование файла /etc/passwd (и, возможно, /etc/shadow) с помощью редактора vipw
- редактирование файла /etc/group с помощью vigr, если нужна новая группа
- Создание домашнего каталога mkdir
- Копирование "скелета" /etc/skel в домашний каталог
- Установка атрибутов для домашнего каталога и его содержимого с помощью chown и chmod с ключом -R:
  - cd /home/ivanov
  - chown -R ivanov.users .
  - chmod -R u+rwX, go= .
  - chmod go=x .
- Установка пароля с помощью passwd
- Задание персональной информации и шелла: **chfn, chsh**
- Удаление – обратная задача. Найти файлы: find / -user ivanov
- Временное блокирование: chsh –s /usr/local/lib/no-login/security ivanov

Архивирование, аудит

- tar, gzip, etc.
- SYSLOG

Архивирование: tar, cpio, dump

- Простейшее архивирование (утилиты BSD tar и AT&T cpio) :
  - tar –cvzf /dev/ftape /home tar –czf 123.tgz ./ ls | cpio –o >arhive.cpio
- Восстановление «тарбалла»
  - tar –xvzf /dev/ftape /
- Многоуровневое архивирование
  - Два типа: полное и изменений (инкрементное)
  - Для 10 лент можно определить, например, 3-уровневое А:
    - 2 ленты 0 уровня (первая пятница месяца, полное архивирование)
    - 4 ленты 1 уровня (1 раз в неделю, пятница , архивирование изменений. В месяце может быть 5 пятниц.)
    - 4 ленты 2 уровня (понедельник, вторник, среда и четверг, архивирование изменений)



Службы наблюдения, протоколирования. Система регистрации событий SYSLOG

- В ОС UNIX нет встроенной системы аудита, а есть регистратор событий SYSLOG (XPG4-UNIX, CISCO)
  - facility – система, источник сообщения
    - kern, user, mail, daemon, auth, local0 .. local7
    - соответствующие константы LOG_KERN, ...
  - priority/level – важность сообщения
    - emerg, alert, crit, err, warning, notice, info, debug
    - соответствующие константы: LOG_EMERG, ...
  - Использует дейтаграммный транспорт – UDP/IP, порт 514 (при запуске демона syslogd в сетевом режиме: syslogd -r)

Библиотечные функции системы SYSLOG

```
openlog(const char *id, int log_option, int facility);
```

Типичное использование:

```
openlog(p->myname,POP_LOGOPTS,POP_FACILITY);
syslog (stat,"%s",msgbuf);
closelog();
```

Функции:

openlog(), closelog() открытие/инициализация лога и закрытие
syslog() - запись в **syslog**
setlogmask() – установка битов, разрешающих протоколирование для заданных приоритетов. Установленный бит – протоколирование сообщение с данным приоритетом разрешено.

Конфигурация SYSLOG

```
#
# Конфигурационный файл демона syslogd
#
kern.debug              /var/adm/syslog/kern.log
kern.debug              /dev/console
daemon.debug            /var/adm/syslog/daemon.log
auth.debug              /var/adm/syslog/auth.log
syslog.debug            /var/adm/syslog/syslog.log
*.notice;mail.info      /var/adm/syslog/mail
*.crit                  /var/adm/syslog/critical
kern.err                @nix.cs.vsu.ru
*.emerg                 *
*.alert                 andrey, sergey
*.alert;auth.warning    ivan
```

Недостатки и достоинства SYSLOG

- Некоторые недостатки (некоторые устранены в последующих вариантах, например, syslog-ng):
  - события поступают от приложений и проверить правильность их нельзя;
  - нельзя проверить источник сообщения;
  - тип сообщения определяет приложение;
  - нет защиты от привилегированного приложения/пользователя, что частично компенсируется немодифицируемостью вывода (например, печать)
- Некоторые достоинства:
  - поддерживает прием сообщений по сети;
  - возможно создание изолированного сервера-регистратора;
  - поддерживается сетевым оборудованием: маршрутизаторы, управляемые коммутаторы, точки доступа беспроводных сетей и т.п.