

Заметки о файловой системе MINIX3

Цилюрик О.И.

< olej@front.ru >

Редакция 2.12

от 23.01.2010

Оглавление

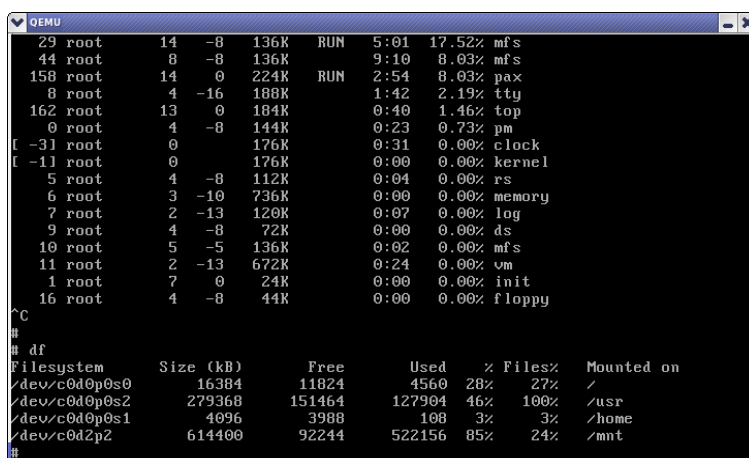
Аннотация.....	1
Введение.....	1
Автономное создание разделов.....	2
Подразделы (диска).....	6
Переразметка.....	6
Инсталляция X11R6	6
Минимальная установка.....	8
RAM-диск.....	11
Источники информации.....	13

Аннотация

В следующем ниже тексте описываются «ручные» манипуляции с разделами дисков в операционной системе MINIX3. Под «ручными» манипуляциями предполагается, что разделы диска создаются не в процессе начальной установки системы (в ходе экранного диалога), а консольными командами уже установленной системы. Затрагиваются, в том числе, и вопросы создания и использования RAM-дисков.

Введение

Первоначальным побудительным мотивом написания данных заметок стала попытка установки MINIX3 с X11R6 в дисковое пространство **минимального** объёма, в ходе отработки приёмов удалённой работы с системой по сети, когда из установленного программного обеспечения в системе требовалось иметь всего только **бинарные** установки X11R6 и SSL/SSH. Установка производилась в среде виртуализации QEMU, но установка и в реальный раздел диска не имеет в этой части никаких отличий. Некоторой неприятной неожиданностью стало то, что такую инсталляцию не удавалось произвести в диск размером в 300Mb (при заявленной минимальной потребности MINIX3 в 79Mb). Установка завершается с аварийным сообщением: *out of i-nodes on device 3/134*; а вот команда `df` говорит следующее:



```
QEMU
29 root 14 -8 136K RUN 5:01 17.52% mfs
44 root 8 -8 136K 9:10 8.03% mfs
158 root 14 0 224K RUN 2:54 8.03% pax
8 root 4 -16 188K 1:42 2.19% tty
162 root 13 0 184K 0:40 1.46% top
0 root 4 -8 144K 0:23 0.73% pm
[-31 root 0 176K 0:31 0.00% clock
[-11 root 0 176K 0:00 0.00% kernel
5 root 4 -8 112K 0:04 0.00% rs
6 root 3 -10 736K 0:00 0.00% memory
7 root 2 -13 120K 0:07 0.00% log
9 root 4 -8 72K 0:00 0.00% ds
10 root 5 -5 136K 0:02 0.00% mfs
11 root 2 -13 672K 0:24 0.00% vm
1 root 7 0 24K 0:00 0.00% init
16 root 4 -8 44K 0:00 0.00% floppy
^C
#
# df
#
Filesystem      Size (kB)    Free      Used      % Files%    Mounted on
/dev/c0d0p0s0   16384       11824     4560     28%    27%    /
/dev/c0d0p0s2  279368     151464   127904   46%   100%    /usr
/dev/c0d0p0s1   4096        3988     108      3%     3%     /home
/dev/c0d2p2    614400     92244   522156  85%   24%    /mnt
#
```

Рис.0.1

На разделе (*partition*) `/dev/c0d0p0s2` файловой системы — более 151Mb свободного пространства (больше 50% общего объёма раздела), но исчерпалось число записей о файлах в таблицах *i*-узлов (*i-node*); ничего больше в этом разделе создать нельзя.

В следующем ниже тексте мы найдём способы разрешить подобные «казусы», а, попутно, рассмотрим работу команд MINIX3, относящихся к её файловой системе.

Автономное создание разделов

В этой части мы создадим достаточно большой (порядка 1Gb) диск (виртуальный) для MINIX3, но, загрузившись с LiveCD (или с ранее инсталлированного MINIX3 виртуального диска), не станем, как обычно, на этот диск устанавливать систему, а займёмся автономной (ручной) разбивкой диска на разделы, его монтированием... — фактически всем тем, что делает система при инсталляции с установочного CD. Создадим (в Linux) образ такого диска:

```
$ dd of=minix3-disk bs=1024 seek=1000000 count=0
0+0 записей считано
0+0 записей написано
скопировано 0 байт (0 B), 0,00382674 секунд, 0,0 kB/s
```

```
$ ls -l minix3-disk
```

```
-rw-rw-r-- 1 olej olej 1024000000 Ноя 15 09:46 minix3-disk
```

Загружаем систему с LiveCD:

```
$ qemu -m 50M -hda minix3-disk -cdrom /dev/cdrom -boot d -localtime -kernel-kqemu
...
```

Прежде всего, командой:

```
# repartition /dev/c0d0
```

```
- создаём пустую запись MBR на диске /dev/c0d0.
```

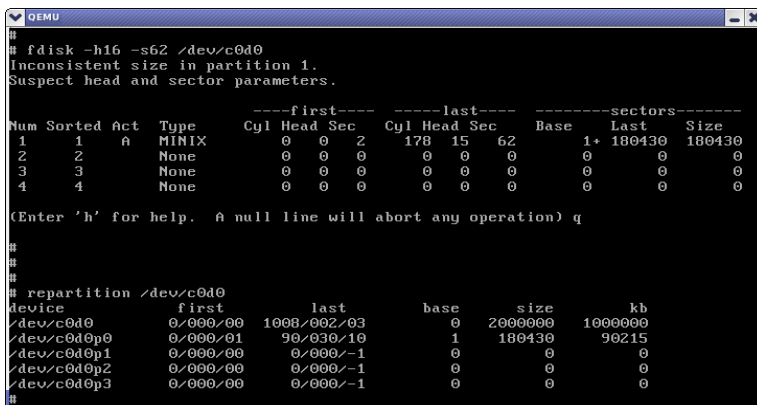


Рис.1.1

Пространство диска, созданное dd (в таком как показано ранее синтаксисе) содержит мусор, прежде всего делам repartition (создание MBR); значения в разделе /dev/c0d0p0 — это следы мусора от предыдущего файла виртуального диска, размещавшегося на этом же месте.

Далее разбиваем диск на требуемое нам число разделов, в соответствии с требуемыми (планируемыми) размерами разделов, например так:

```
# partition /dev/c0d0 81:2000 81:10000 81:50000 81:0+
```

(4 раздела типа MINIX3 — код раздела 81, размерами, соответственно, 1, 5, и 25 Mb, размеры указаны в 512-байт секторах; 4-му разделу остаётся всё остальное доступное на диске пространство).

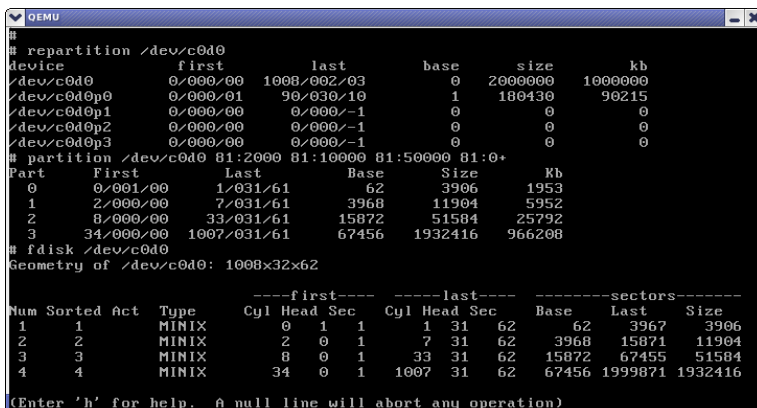
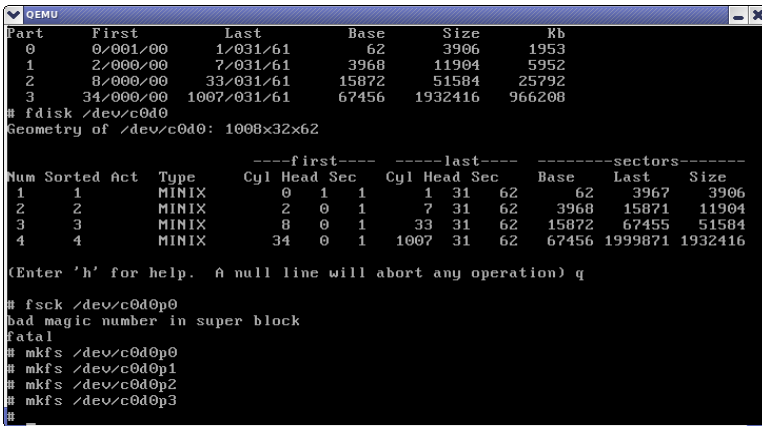


Рис.1.2

Создаём (команда partition) одновременно 4 первичных (primary) раздела, параметрами для partition «заказаны» размеры разделов: 1Mb, 5Mb, 25Mb и остаток от полного размера диска. Получилось в итоге: 2Mb, 6Mb, 26Mb и остаток 966Mb.

В диалоговом режиме, в точности того же результата в создании разделов (или чуть более точно контролируемого процесса в диалоге) можно было бы добиться в результате выполнения команды:

```
# fdisk /dev/c0d0
```



```
QEMU
Part  First      Last      Base      Size      Kb
0    0/001/00    1/031/61     62      3906     1953
1    2/000/00    7/031/61    3968     11904    5952
2    8/000/00   33/031/61   15872    51584   25792
3   34/000/00  1007/031/61 67456   1932416 966208
# fdisk /dev/c0d0
Geometry of /dev/c0d0: 1008x32x62

-----first-----last-----sectors-----
Num Sorted Act  Type      Cyl Head Sec  Cyl Head Sec  Base  Last  Size
1    1    1  MINIX      0   1   1    1  31  62    62   3967  3906
2    2    2  MINIX      2   0   1    7  31  62   3968  15871 11904
3    3    3  MINIX      8   0   1   33  31  62   15872 67455 51584
4    4    4  MINIX     34   0   1  1007 31  62   67456 1932416 966208

(Enter 'h' for help.  A null line will abort any operation) q
# fsck /dev/c0d0p0
bad magic number in super block
fatal
# mkfs /dev/c0d0p0
# mkfs /dev/c0d0p1
# mkfs /dev/c0d0p2
# mkfs /dev/c0d0p3
```

Рис.1.3

К этому времени в самих разделах всё ещё находится мусор (не форматированы), на это отчётливо указывает результат выполнения команд типа:

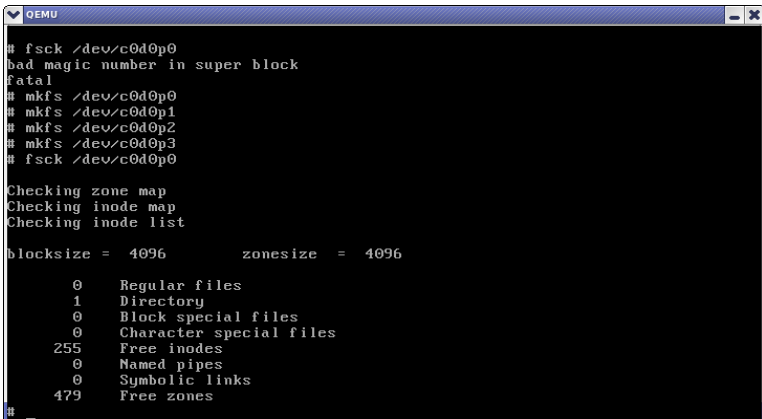
```
# fsck /dev/c0d0p0
```

— результат которой я показываю, для убедительности, прежде выполнения форматирования командами вида:

```
# mkfs /dev/c0d0p0
```

Выполняем `mkfs` для всех 4-х разделов, при этом (пока) выполняем `mkfs` по умолчанию (не заказываем ему ни размер блока, ни размер таблицы i-узлов).

Теперь детально рассматриваем (повторно командой `fsck`) полученные разделы (отличающиеся на порядки, по величине), которые мы создали...



```
QEMU
# fsck /dev/c0d0p0
bad magic number in super block
fatal
# mkfs /dev/c0d0p0
# mkfs /dev/c0d0p1
# mkfs /dev/c0d0p2
# mkfs /dev/c0d0p3
# fsck /dev/c0d0p0

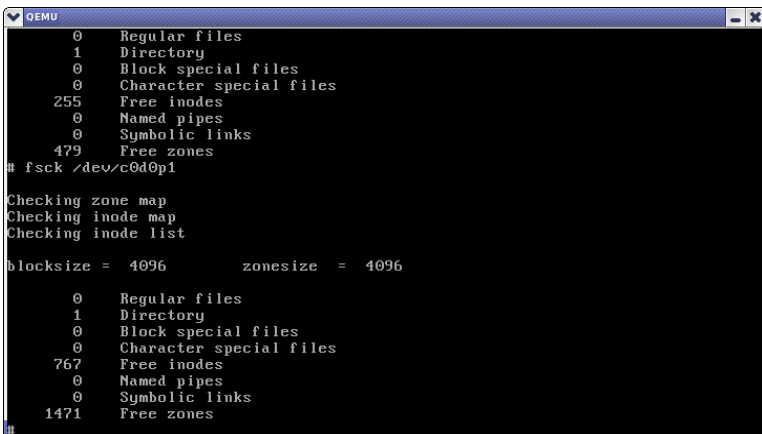
Checking zone map
Checking inode map
Checking inode list

blocksize = 4096      zonesize = 4096

    0  Regular files
    1  Directory
    0  Block special files
    0  Character special files
   255 Free inodes
    0  Named pipes
    0  Symbolic links
   479 Free zones
```

Рис.1.4

/dev/c0d0p0 2Mb раздел — 255 i-узлов...



```
QEMU
    0  Regular files
    1  Directory
    0  Block special files
    0  Character special files
   255 Free inodes
    0  Named pipes
    0  Symbolic links
   479 Free zones
# fsck /dev/c0d0p1

Checking zone map
Checking inode map
Checking inode list

blocksize = 4096      zonesize = 4096

    0  Regular files
    1  Directory
    0  Block special files
    0  Character special files
   767 Free inodes
    0  Named pipes
    0  Symbolic links
  1471 Free zones
```

Рис.1.5

/dev/c0d0p1 6Mb раздел — 767 i-узлов...

```

QEMU
0 Regular files
1 Directory
0 Block special files
0 Character special files
767 Free inodes
0 Named pipes
0 Symbolic links
1471 Free zones
# fsck /dev/c0d0p2

Checking zone map
Checking inode map
Checking inode list

blocksize = 4096      zonesize = 4096

0 Regular files
1 Directory
0 Block special files
0 Character special files
3263 Free inodes
0 Named pipes
0 Symbolic links
6392 Free zones
#

```

Рис.1.6

/dev/c0d0p2 26Mb раздел — 3263 i-узлов...

```

QEMU
0 Regular files
1 Directory
0 Block special files
0 Character special files
3263 Free inodes
0 Named pipes
0 Symbolic links
6392 Free zones
# fsck /dev/c0d0p3

Checking zone map
Checking inode map
Checking inode list

blocksize = 4096      zonesize = 4096

0 Regular files
1 Directory
0 Block special files
0 Character special files
34559 Free inodes
0 Named pipes
0 Symbolic links
240999 Free zones
#

```

Рис.1.7

/dev/c0d0p3 966Mb раздел — 34559 i-узлов...

Полученные таким образом каждый из разделов диска (*partition*) мы теперь можем монтировать к произвольной точке дерева «/» файловой системы MINIX3, создавать там произвольно каталоги и файлы, т.е. **произвольным образом расширять дерево «/» файловой системы**, например так:

```
# mkdir xxx
# mount /dev/c0d0p0 /xxx
```

(монтирование без указания типа файловой системы предполагает, что тип её — MINIX3).

```

QEMU
#
#
#
#
#
# pwd
# /root
# mkdir /xxx
# mount /dev/c0d0p0 /xxx
# /dev/c0d0p0 is read-write mounted on /xxx
# cd /xxx
#
# echo 1234567 > yyyy
# touch zzzz
#
# ls -l
total 1
-rw-r--r-- 1 root operator 8 Nov 15 11:43 yyyy
-rw-r--r-- 1 root operator 0 Nov 15 11:43 zzzz
#

```

Рис.1.8

Вернёмся коротко к вопросу инсталляции пакета X11R6 в дисковый раздел, с которого началось всё рассмотрение. Инсталляция, в терминах MINIX3, по крайней мере в отношении бинарных пакетов, которая нас и интересует — это не что иное, как просто разархивирование *.tar.bz2 архивов в надлежащие места файловой системы. Поэтому мы можем симитировать инсталляцию операциями с архивами и оценить объём этих архивов (рис.1.9-1.11). На этом рисунке крайне любопытно видеть причину, по которой раздел установочного LiveCD /dev/c0d2p2, содержащий пакеты ПО, не удаётся примонтировать к файловой системе /dev/ram : busy — он уже примонтирован после загрузки LiveCD как /usr/install/packages.

```

DEMU
# mount: Can't mount /dev/d0c0p3 on /X11: Bad file number
# mount /dev/d0c0p2 /X11
# mount: Can't mount /dev/d0c0p2 on /X11: Bad file number
# umount /dev/xxx
# umount: Bad file number
# mount /dev/c0d0p0
/dev/c0d0p0 unmounted from /xxx
# mount /dev/c0d0p2 /xxx
/dev/c0d0p2 is read-write mounted on /xxx
# pwd
/usr/install/packages
# ls -l X11*
-rw-r--r-- 1 root operator 96484502 Nov  5 09:59 X11R6.8.2.tar.bz2
# umount /dev/c0d0p2
/dev/c0d0p2 unmounted from /xxx
# mount /dev/c0d0p3 /xxx
/dev/c0d0p3 is read-write mounted on /xxx
# time cp X11* /xxx
11:23.06 real    4.86 user    8:59.16 sys
# bzip2 -d X11*
bzip2: Can't create output file X11R6.8.2.tar: Read-only file system.
# cd /xxx
# time bzip2 -d X11*
37:15.01 real  12:18.21 user  19:25.65 sys
#

```

Рис.1.9

```

DEMU
Checking zone map
Checking inode map
Checking inode list
blocksize = 4096      zonesize = 4096
3 Regular files
2 Directories
0 Block special files
0 Character special files
34555 Free inodes
0 Named pipes
0 Symbolic links
191651 Free zones
# time tar -tof X11R6* > X11.lst
1:03:30.60 real  6:12.33 user  56:42.30 sys
# ls -l
total 197866
-rw-r--r-- 1 root operator 60 Jan 13 2009 .minixpackage
-rwxr-xr-x 1 root operator 626 Dec 10 2008 .postinstall
drwxr-xr-x 2 root operator 128 Nov 15 19:38 X11
-rw-r--r-- 1 root operator 698935 Nov 15 22:19 X11.lst
-rw-r--r-- 1 root operator 201912320 Nov 15 12:24 X11R6.8.2.tar
#

```

Рис.1.10

```

DEMU
# mkdir xxx
# pwd
/root
# ls
.ashrc .ellepro.b1 .ellepro.e .exerc .profile xxx
# rm xxx
rm: xxx: Is a directory
# rmdir xxx
#
# cd /
# mkdir xxx
# mount /dev/c0d0p3 /xxx
/dev/c0d0p3 is read-write mounted on /xxx
# ls /xxx
.minixpackage .postinstall X11 X11.lst X11R6.8.2.tar
# ls -l /xxx
total 197866
-rw-r--r-- 1 root operator 60 Jan 13 2009 .minixpackage
-rwxr-xr-x 1 root operator 626 Dec 10 2008 .postinstall
drwxr-xr-x 2 root operator 128 Nov 15 19:38 X11
-rw-r--r-- 1 root operator 698935 Nov 15 22:19 X11.lst
-rw-r--r-- 1 root operator 201912320 Nov 15 12:24 X11R6.8.2.tar
# wc -l /xxx/X11.lst
10728 /xxx/X11.lst
#

```

Рис.1.11

Резльтирующая оценка: 10728 каталогов и файлов — суммарный размер таблицы i-узлов, необходимый для бинарной установки X11R6.

Примечание: выполняя операции с архивами мы попутно оценили и времена, требуемые для выполнения операций над пакетом X11R6: bzip2 — 37 мин.; tar — 63 мин. Хотя примеры этого раздела выполнены на относительно не быстром процессоре (показана команда Linux!):

```

$ cat /proc/cpuinfo
processor      : 0
vendor_id    : GenuineIntel
cpu family   : 6
model        : 8
model name   : Celeron (Coppermine)
stepping     : 3
cpu MHz      : 534.575
cache size   : 128 KB
...

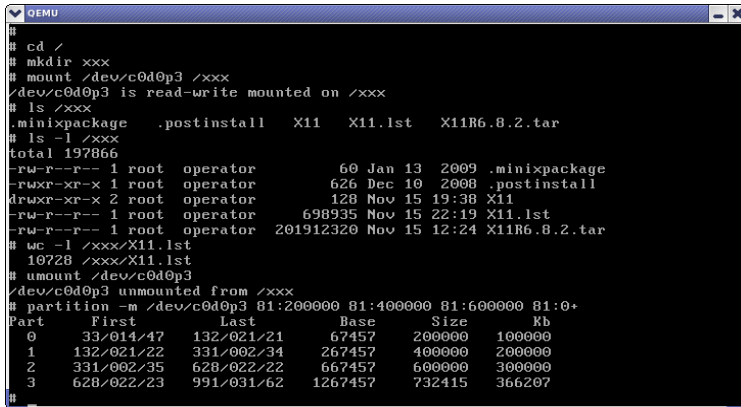
```

Но такие «чудовищные» времена исполнения — это плата за виртуальную машину QEMU (и удобства отработси

под QEMU), что нужно всегда держать в уме! К счастью, все прочие пакеты с LiveCD — **на порядки** меньше, чем X11R6.

Подразделы (диска)

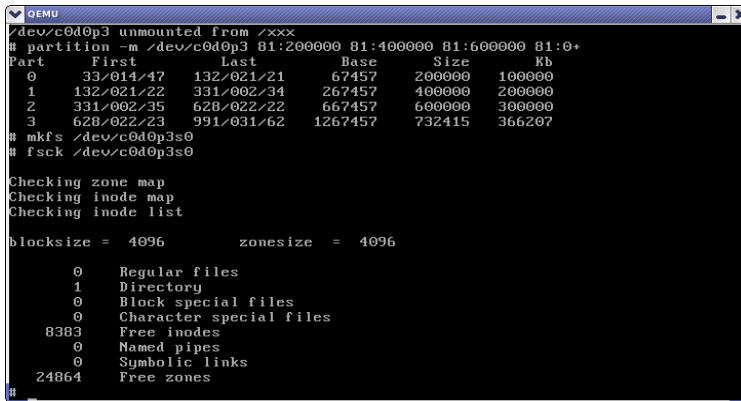
В принципе, на том, до чего мы дошли выше, можно было бы и остановиться. Но MINIX3, подобно некоторым другим UNIX-подобным системам (SunSolaris/OpenSolaris, FreeBSD) — допускает дальнейшее разбиение разделов (*partition*) на подразделы (*subpartition*), в некоторых операционных системах для этого понятия ещё используется термин *slices*.



```
QEMU
#
# cd /
# mkdir xxx
# mount /dev/c0d0p3 /xxx
/dev/c0d0p3 is read-write mounted on /xxx
# ls /xxx
.minixpackage .postinstall X11 X11.lst X11R6.8.2.tar
# ls -l /xxx
total 197866
-rw-r--r-- 1 root operator      60 Jan 13  2009 .minixpackage
-rwxr-xr-x 1 root operator     626 Dec 10  2008 .postinstall
drwxr-xr-x 2 root operator     128 Nov 15 19:38 X11
-rw-r--r-- 1 root operator    698935 Nov 15 22:19 X11.lst
-rw-r--r-- 1 root operator   201912320 Nov 15 12:24 X11R6.8.2.tar
# wc -l /xxx/X11.lst
10728 /xxx/X11.lst
# umount /dev/c0d0p3
/dev/c0d0p3 unmounted from /xxx
# partition -m /dev/c0d0p3 81:200000 81:400000 81:600000 81:0+
Part  First      Last      Base      Size      Kb
0    33/014/47   132/021/21 67457    200000    100000
1    132/021/22  331/002/34 267457    400000    200000
2    331/002/35  628/022/22 667457    600000    300000
3    628/022/23  991/031/62 1267457   732415    366207
```

Рис.1.12

Точно так же, как диск /dev/c0d0 мы поделили на 4 раздела — мы делим раздел /dev/c0d0p3 на 4 подраздел (*subpartition*) размером: 100Mb, 200Mb, 300Mb и остаток пространства раздела — 366Mb.



```
QEMU
/dev/c0d0p3 unmounted from /xxx
# partition -m /dev/c0d0p3 81:200000 81:400000 81:600000 81:0+
Part  First      Last      Base      Size      Kb
0    33/014/47   132/021/21 67457    200000    100000
1    132/021/22  331/002/34 267457    400000    200000
2    331/002/35  628/022/22 667457    600000    300000
3    628/022/23  991/031/62 1267457   732415    366207
# mkfs /dev/c0d0p3s0
# fsck /dev/c0d0p3s0

Checking zone map
Checking inode map
Checking inode list

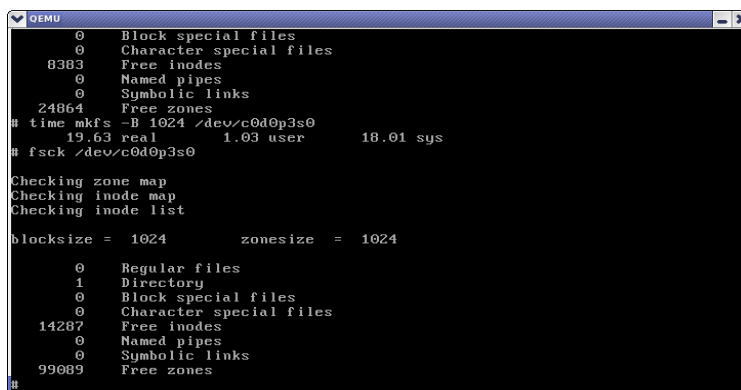
blocksize = 4096      zonesize = 4096

0    Regular files
1    Directory
0    Block special files
0    Character special files
8383 Free inodes
0    Named pipes
0    Symbolic links
24864 Free zones
```

Рис.1.13

Разметку (форматирование) подраздела делаем в точности так, как и для раздела, с помощью `mkfs`. Показано состояние подраздела /dev/c0d0p3s0, размером 100Mb.

Переразметка



```
QEMU
0    Block special files
0    Character special files
8383 Free inodes
0    Named pipes
0    Symbolic links
24864 Free zones
# time mkfs -B 1024 /dev/c0d0p3s0
19.63 real    1.03 user    18.01 sys
# fsck /dev/c0d0p3s0

Checking zone map
Checking inode map
Checking inode list

blocksize = 1024      zonesize = 1024

0    Regular files
1    Directory
0    Block special files
0    Character special files
14287 Free inodes
0    Named pipes
0    Symbolic links
99089 Free zones
```

Рис.1.14

Форматирование того же подраздела (100Mb), но с размером блока не 4Kb (по умолчанию), а принудительно установленным в 1Kb. Размер таблицы i-узлов увеличился практически в 2 (!?) раза.

Инсталляция X11R6

Теперь, отвлекаясь чуть в сторону, пройдем тот же путь рассмотрения файловой системы MINIX3, но несколько другим образом: установим образ OS в заведомо достаточное пространство диска (4Gb), и установим ПО X11R6 (в достаточное для него пространство), контролируя состояние файловой системы на каждом шаге.

```
$ dd of=minix3-disk bs=1024 seek=4000000 count=0
```

```
0+0 записей считано
```

```
0+0 записей написано
```

скопировано 0 байт (0 В), 1,8893e-05 секунд, 0,0 kB/s

```
$ ls -l
```

итого 626280

```
-rw-rw-r-- 1 olej olej 4096000000 Ноя 16 11:30 minix3-disk
```

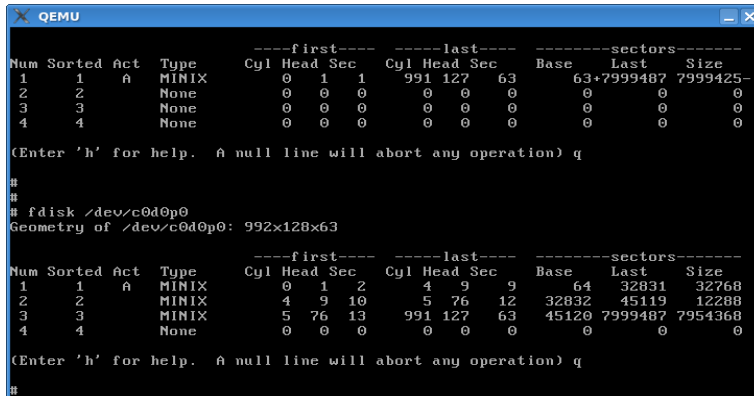
```
-rw-rw-r-- 1 olej olej 640679936 Ноя 6 10:25 minix_R3.1.5-r5612.iso
```

Загрузка с LiveCD:

```
$ qemu -cdrom /home/download/MINIX3/minix_R3.1.5-r5612.iso -hda minix3-disk \
```

```
-boot d -kernel-kqemu
```

...



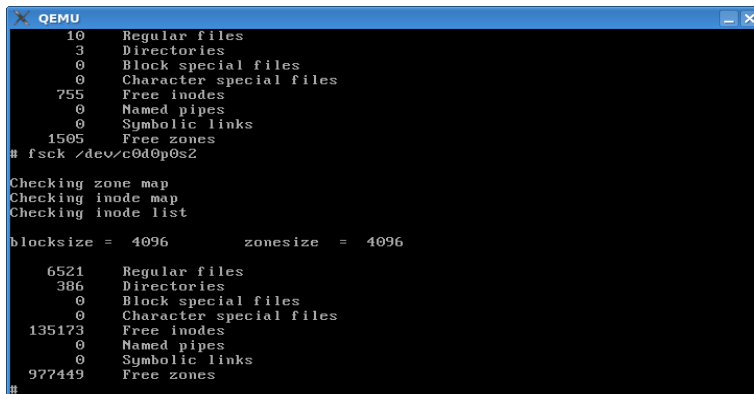
```
QEMU
-----first-----last-----sectors-----
Num Sorted Act Type Cyl Head Sec Cyl Head Sec Base Last Size
1 1 A MINIX 0 1 1 991 127 63 63+7999487 7999425-
2 2 None 0 0 0 0 0 0 0 0 0
3 3 None 0 0 0 0 0 0 0 0 0
4 4 None 0 0 0 0 0 0 0 0 0
(Enter 'h' for help. A null line will abort any operation) q
#
#
# fdisk /dev/c0d0p0
Geometry of /dev/c0d0p0: 992x128x63
-----first-----last-----sectors-----
Num Sorted Act Type Cyl Head Sec Cyl Head Sec Base Last Size
1 1 A MINIX 0 1 2 4 9 9 64 32831 32768
2 2 MINIX 4 9 10 5 76 12 32832 45119 12288
3 3 MINIX 5 76 13 991 127 63 45120 7999487 7954368
4 4 None 0 0 0 0 0 0 0 0 0
(Enter 'h' for help. A null line will abort any operation) q
#
```

Рис.2.1

Выполняем установку системы:

```
# setup
```

Так выглядит карта разделов диска сразу после установки MINIX3 в раздел /dev/c0d0p0.



```
QEMU
10 Regular files
3 Directories
0 Block special files
0 Character special files
755 Free inodes
0 Named pipes
0 Symbolic links
1505 Free zones
# fsck /dev/c0d0p0s2
Checking zone map
Checking inode map
Checking inode list
blocksize = 4096 zonesize = 4096
6521 Regular files
386 Directories
0 Block special files
0 Character special files
135173 Free inodes
0 Named pipes
0 Symbolic links
977449 Free zones
#
```

Рис.2.2

Так создан раздел /dev/c0d0p0s2 (/usr) сразу же после установки системы.

Выполняем:

```
# shutdown now
```

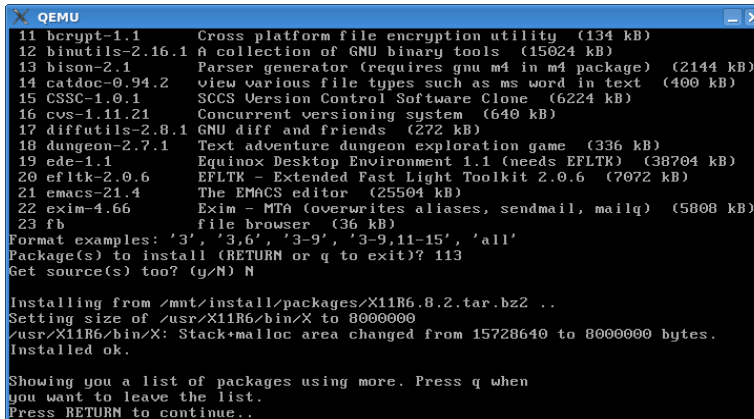
```
> off
```

Перезагружаем систему, но теперь уже с установленного образа на диск /dev/c0d0p0 :

```
$ qemu -cdrom /home/download/MINIX3/minix_R3.1.5-r5612.iso -hda minix3-disk \
```

```
-boot c -kernel-kqemu
```

...



```
QEMU
11 bcrpyt-1.1 Cross platform file encryption utility (134 kB)
12 binutils-2.16.1 A collection of GNU binary tools (15024 kB)
13 bison-2.1 Parser generator (requires gnu m4 in m4 package) (2144 kB)
14 catdoc-0.94.2 view various file types such as ms word in text (400 kB)
15 CSSC-1.0.1 SCCS Version Control Software Clone (6224 kB)
16 cvs-1.11.21 Concurrent versioning system (640 kB)
17 diffutils-2.8.1 GNU diff and friends (272 kB)
18 dungeon-2.7.1 Text adventure dungeon exploration game (336 kB)
19 ede-1.1 Equinox Desktop Environment 1.1 (needs EFLTK) (38704 kB)
20 efltk-2.0.6 EFLTK - Extended Fast Light Toolkit 2.0.6 (7072 kB)
21 emacs-21.4 The Emacs editor (25504 kB)
22 exim-4.66 Exim - MTA (overwrites aliases, sendmail, mailq) (5808 kB)
23 fb file browser (36 kB)
Format examples: '3', '3.6', '3-9', '3-9,11-15', 'all'
Package(s) to install (RETURN or q to exit)? 113
Get source(s) too? (y/N) N
Installing from /mnt/install/packages/X11R6.8.2.tar.bz2 ..
Setting size of /usr/X11R6/bin/X to 8000000
/usr/X11R6/bin/X: Stack+malloc area changed from 15728640 to 8000000 bytes.
Installed ok.
Showing you a list of packages using more. Press q when
you want to leave the list.
Press RETURN to continue..
```

Рис.2.3

После загрузки системы и регистрации как root, выполнили:

```
# packman
```

Выбрали для установки X11R6 (пакет номер 113). Показано завершение установки.

```

#
# df
#
Filesystem      Size (kB)    Free      Used    % Files%  Mounted on
/dev/c0d0p0s0    16384        11740     4644    29%    27%    /
/dev/c0d0p0s2   3977184     3688992   288192  8%     13%    /usr
/dev/c0d0p0s1     6144         6020      124     3%     2%     /home
/dev/c0d2p2     614400      92244     522156 85%    24%    /mnt
# fsck /dev/c0d0p0s2

Checking zone map
Checking inode map
Checking inode list

blocksize = 4096      zonesize = 4096

15466 Regular files
  713 Directories
    0 Block special files
    0 Character special files
124451 Free inodes
    0 Named pipes
  1450 Symbolic links
 922248 Free zones
#

```

Рис.2.4

А так выглядит уже показанный подраздел /dev/c0d0p0s2 (/usr), но сразу после установки X11R6:

135173 — 124451 = 10722

- столько i-узлов «отъела» установка X11R6, это очень хорошо сочетается с оценкой, полученной ранее совсем другим путём.

Минимальная установка

Задача: теперь предыдущую установку на диск 4Gb, зная оценки потребностей пакетов – повторить и установить систему в виртуальный диск минимального объёма, который (файл виртуального диска) можно носить с собой на USB flash и т.п. Будем устанавливать **только бинарные** пакеты, и в дополнение к X11R6 установим SSL/SSH (для обеспечения удалённого доступа к MINIX3 по виртуальной сети QEMU).

Создаём виртуальный диск размером ~300Mb (далее станет видно, что меньше – мало, а больше – это переизбыток, и таким образом это минимальный размер, в который можно установить MINIX3 + X11R6, и то с некоторыми ухищрениями, о которых дальше...):

```

$ time dd if=/dev/zero of=minix3-300 bs=1k count=300k
307200+0 записей считано
307200+0 записей написано
   скопировано 314572800 байт (315 MB), 2,25056 секунд, 140 MB/s
real    0m2.351s
user    0m0.168s
sys     0m1.948s

```

Обратите внимание, что диск создаётся командой dd в другом синтаксисе, чем ранее ... для разнообразия, но это по-другому (по времени) и работает.

Загрузка с LiveCD и установка MINIX3 на виртуальный HDD:

```

$ qemu -hda minix3-300 -cdrom minix_r3.1.5-r5612.iso -boot d -localtime -kernel-kqemu

```

```

# QEMU
Unless you are an expert, you are advised to use the automated
step-by-step help in setting up.

Press ENTER for automatic mode, or type 'expert':

--- Substep 3.1: Select a disk to install MINIX 3 -----
Probing for disks. This may take a short while..... Probing done.
The following disk was found on your system:

  Disk [0]: /dev/c0d0, 299 MB
           Free space ( 299 MB)

Enter the disk number to use: [0] 0

--- Substep 3.2: Select a disk region -----
Please select the region that you want to use for the MINIX 3 setup.
If you select an in-use region it will be overwritten by MINIX. The
following region were found on the selected disk:

  [0] Free space ( 299 MB)

Enter the region number to use or type 'delete': [0] 0

```

Рис.2.5

После загрузки:

```
# setup
```

При определении HDD я даже не стал входить в режим «expert» – всё происходит автоматически.


```

QEMU
This is the point of no return. You have selected to install MINIX 3
into region 0 of disk /dev/c0d0. Please confirm that you want
to use this selection to install MINIX 3.

Are you sure you want to continue? Please enter 'yes' or 'no': yes

--- Step 4: Reinstall choice -----
No old /home found. Doing full install.

--- Step 5: Select the size of /home -----
MINIX will take up 79 MB, without /home.
How big do you want your /home to be in MB (0-219) ? [43] 4
4 MB Ok? [Y] y

--- Step 6: Select a block size -----
The default file system block size is 4 KB.
Block size in kilobytes? [4] 1

```

Рис.2.6

Вот здесь – важнейший трюк (который я выше упомянул как «ухищрение»): вместо умолчания размера блока 4Kb, я определяю 1Kb. Если этого не сделать, вы не сможете установить X в HDD объёмом менее ~1.2Gb!

```

QEMU
by MINIX are marked with *. Please choose:

0. No Ethernet card (no networking)
1. Intel Pro/100
2. 3Com 501 or 3Com 509 based card
3. Realtek 8139 based card (also emulated by KVM)
4. * Realtek 8029 based card (also emulated by Qemu)
5. NE2000, 3com 503 or WD based card (also emulated by Bochs)
6. AMD LANCE (also emulated by VMware and VirtualBox)
7. Different Ethernet card (no networking)

Ethernet card? [4]

Configure network using DHCP or manually?

1. Automatically using DHCP
2. Manually

Configure method? [1] 2
Hostname [minix]: qemu-minix
IP address [10.0.0.11]: 192.168.3.4
Netmask [255.255.255.0]:
Gateway: 192.168.3.6
Primary DNS Server [1]: 192.168.9.2
Secondary DNS Server [1]: 192.168.9.254

```

Рис.2.7

Конфигурируем сеть. Если мы «угадаем» параметры здесь, на этапе установки системы, то нам не придётся настраивать сеть вручную. Виртуальную сеть будем строить на IP=192.168.3.4 со стороны гостевого MINIX3, и IP=192.168.3.6 со стороны базового Linux.

Всё, установка с LiveCD закончилась, можно переходить к установке пакетов, для этого перегружаемся с HDD:

```
$ qemu -hda minix3-300 -cdrom minix_R3.1.5-r5612.iso -boot c -localtime -kernel-kqemu
```

...

```

QEMU
11 bcrpyt-1.1 Cross platform file encryption utility (134 kB)
12 binutils-2.16.1 A collection of GNU binary tools (15024 kB)
13 bison-2.1 Parser generator (requires gnu m4 in m4 package) (2144 kB)
14 catdoc-0.94.2 view various file types such as ms word in text (400 kB)
15 c33c-1.0.1 SCS Version Control Software Clone (6224 kB)
16 cvs-1.11.21 Concurrent Versioning System (640 kB)
17 diffutils-2.8.1 GNU diff and friends (272 kB)
18 dungeon-2.7.1 Text adventure dungeon exploration game (336 kB)
19 ede-1.1 Equinox Desktop Environment 1.1 (needs EFLTK) (38704 kB)
20 efltk-2.0.6 EFLTK - Extended Fast Light Toolkit 2.0.6 (7072 kB)
21 emacs-21.4 The EMACS editor (25504 kB)
22 exim-4.66 Exim - MTA (overwrites aliases, sendmail, mailq) (5808 kB)
23 fb file browser (36 kB)
Format examples: '3', '3.6', '3-9', '3-9,11-15', 'all'
Package(s) to install (RETURN or q to exit)? 113
Get source(s) too? (y/N) N

Installing from /mnt/install/packages/X11R6.8.2.tar.bz2 ..
Setting size of /usr/X11R6/bin/X to 8000000
/usr/X11R6/bin/X: Stack+malloc area changed from 15728640 to 8000000 bytes.
Installed ok.

Showing you a list of packages using more. Press q when
you want to leave the list.
Press RETURN to continue..

```

Рис.2.8

После регистрации как root:

```
# packman
```

- и устанавливаем пакеты X11R6 (#113) и SSL/SSH (#71/72, на рисунках не показаны).

Посмотрим состояние файловой системы (особо /usr) после установки всех нужных нам пакетов:

```

QEMU
# df
Filesystem      Size (kB)    Free      Used    % Files%  Mounted on
/dev/c0d0p0s0  16384       11740     4644    29%    27%    /
/dev/c0d0p0s2  206424     29116    257308  90%    45%    /usr
/dev/c0d0p0s1   4096       3941     155     4%     1%    /home
/dev/c0d2p2    614400    92244    522156  85%    24%    /mnt
# time fsck /dev/c0d0p0s2

Checking zone map
Checking inode map
Checking inode list

blocksize = 1024      zonesize = 1024

15856 Regular files
 734 Directories
 0 Block special files
 0 Character special files
22886 Free inodes
 0 Named pipes
1452 Symbolic links
29116 Free zones
3:06.86 real    4.93 user    3:01.88 sys
#

```

Рис.2.9

Занято 90% дискового пространства /usr и 45% таблицы i-узлов. Обратите внимание на время, которое компьютер (на этот раз уже 2x1.6Ghz) «разгребал» сильно заполненную файловую систему MINIX3.

После этого:

```
# shutdown
```

```
> off
```

К этому времени у нас ещё нет сети: она предварительно настроена со стороны MINIX3, но нет тунеля со стороны Linux. Содержимое файла (файловая система Linux) /etc/qemu-ifup:

```
$ cat qemu-ifup
```

```
#!/bin/sh
```

```
echo ----- tap up -----
```

```
sudo /sbin/ifconfig $1 192.168.3.6
```

- возможно, этот файл /etc/qemu-ifup для скрипта подъёма виртуального интерфейса tap0 вам придётся создать.

Запускаем систему с поддержкой сети (обратите внимание на приглашение # - только с правами root):

```
# qemu -hda minix3-300 -cdrom minix_R3.1.5-r5612.iso -boot c -localtime \
```

```
-kernel-kqemu -net nic,vlan=0 -net tap,vlan=0
```

```
----- tap up -----
```

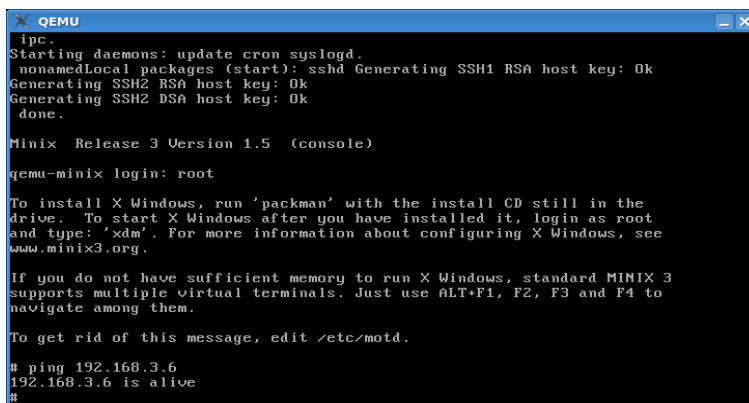
```
...
```

```
# ifconfig tap0
```

```
tap0      Link encap:Ethernet  HWaddr F2:6E:E6:E8:BB:89
          inet addr:192.168.3.6  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fe80::f06e:e6ff:fee8:bb89/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:87 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:420 (420.0 b)  TX bytes:14932 (14.5 KiB)
```

```
$ ping 192.168.3.4
```

```
PING 192.168.3.4 (192.168.3.4) 56(84) bytes of data.
64 bytes from 192.168.3.4: icmp_seq=1 ttl=96 time=15.2 ms
64 bytes from 192.168.3.4: icmp_seq=2 ttl=96 time=6.66 ms
64 bytes from 192.168.3.4: icmp_seq=3 ttl=96 time=6.60 ms
--- 192.168.3.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 6.604/9.518/15.281/4.075 ms
```



```
QEMU
ipc.
Starting daemons: update cron syslogd.
nonamedLocal packages (start): sshd Generating SSH1 RSA host key: Ok
Generating SSH2 RSA host key: Ok
Generating SSH2 DSA host key: Ok
done.
Minix Release 3 Version 1.5 (console)
qemu-minix login: root

To install X Windows, run 'packman' with the install CD still in the
drive. To start X Windows after you have installed it, login as root
and type: 'xdm'. For more information about configuring X Windows, see
www.minix3.org.

If you do not have sufficient memory to run X Windows, standard MINIX 3
supports multiple virtual terminals. Just use ALT+F1, F2, F3 and F4 to
navigate among them.

To get rid of this message, edit /etc/motd.

# ping 192.168.3.6
192.168.3.6 is alive
#
```

Рис.2.10

Подключаемся к MINIX3 по SSH:

```
$ ssh -l root 192.168.3.4
```

- видно запуск демонов и генерацию ключей SSH при старте.

Первая проба сети со стороны MINIX3: ping к Linux.

```

QEMU
To install X Windows, run 'packman' with the install CD still in the
drive. To start X Windows after you have installed it, login as root
and type: 'xdm'. For more information about configuring X Windows, see
www.minix3.org.

If you do not have sufficient memory to run X Windows, standard MINIX 3
supports multiple virtual terminals. Just use ALT+F1, F2, F3 and F4 to
navigate among them.

To get rid of this message, edit /etc/motd.

$
$ ssh -l olej 192.168.3.6
The authenticity of host '192.168.3.6 (192.168.3.6)' can't be established.
RSA key fingerprint is 8f:98:44:55:b1:6e:0b:e5:27:a7:87:f9:8a:3e:d5:43.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.3.6' (RSA) to the list of known hosts.
olej@192.168.3.6's password:
Last login: Tue Nov 17 14:42:14 2009 from 192.168.2.118
olej@opos9 ~1$ uname -a
Linux opos9.altrom.lan 2.6.18-53.1.19.e15 #1 SMP Wed May 7 08:20:19 EDT 2008 i686
i686 i386 GNU/Linux
olej@opos9 ~1$ whoami
olej
olej@opos9 ~1$

```

Рис.2.11

Проверки сети с встречной стороны (вызов из MINIX3 базового Linux по SSH):

Дальше нельзя жить с root без пароля, и без дополнительных пользователей (SSH не позволит).

```

QEMU
To get rid of this message, edit /etc/motd.

# ping 192.168.3.6
192.168.3.6 is alive
#
# passwd
Changing the shadow password of root
New password:
Retype password:
# adduser
Usage: adduser user group home-dir
# adduser olej other /home/olej
cpdir /usr/ast /home/olej
chown -R 12:3 /home/olej
echo olej:0:0::: >>/etc/shadow
echo olej:##olej:12:3:olej:/home/olej: >>/etc/passwd

The new user olej has been added to the system. Note that the password,
full name, and shell may be changed with the commands passwd(1), chfn(1),
and chsh(1). The password is now empty, so only console logins are possible.
# passwd olej
Changing the shadow password of olej
New password:
Retype password:
#

```

Рис.2.12

Добавляем пользователя.

```

root@qemu-minix:/root - Shell #2 - Konsole
Сеанс  Правка  Вид  Закладки  Настройка  Справка

The authenticity of host '192.168.3.4 (192.168.3.4)' can't be established.
RSA key fingerprint is c5:c1:b3:f1:c4:fe:46:ce:4c:bc:f8:a3:32:1b:40:5a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.3.4' (RSA) to the list of known hosts.
root@192.168.3.4's password:
Last login: Wed Nov 18 14:14:20 2009

To install X Windows, run 'packman' with the install CD still in the
drive. To start X Windows after you have installed it, login as root
and type: 'xdm'. For more information about configuring X Windows, see
www.minix3.org.

If you do not have sufficient memory to run X Windows, standard MINIX 3
supports multiple virtual terminals. Just use ALT+F1, F2, F3 and F4 to
navigate among them.

To get rid of this message, edit /etc/motd.

# uname -a
Minix qemu-minix 3 1.5 i686
#

```

Рис.2.13

Из Linux в MINIX3 как root...

Вот к этому времени у нас есть установленная и проверенная система X, на этом и закончим нашу минимальную установку – работающая MINIX3 система с X размещена в виртуальном диске размером ~300Мб.

RAM-диск

Штатная загрузка MINIX3 с Live CD использует для модифицируемых частей системы раздел на RAM-диске. При работе под QEMU вдвойне удобно организовывать раздел на RAM-диск, так как в качестве базовой системы обычно используется Linux с очень большими (по сравнению с аппетитами MINIX3) размерами памяти RAM, заметную часть которой можно «перебросить» MINIX3 для использования в качестве RAM-диска.

После того, что мы уже проделали, поставленная задача не представляет сложностей. Но прежде, взглянем¹ на каталог /dev :

```
# ls -l /dev/ram*
```

¹ В отличие от предыдущих разделов, здесь будут показываться не снимки с консоли, а текстовое содержание сеанса SSH подключения к консоли MINIX3.

```
brw----- 1 root kmem 1,  0 Nov  5 10:00 /dev/ram
brw----- 1 root kmem 1,  7 Nov  5 10:00 /dev/ram0
brw----- 1 root kmem 1,  8 Nov  5 10:00 /dev/ram1
brw----- 1 root kmem 1,  9 Nov  5 10:00 /dev/ram2
brw----- 1 root kmem 1, 10 Nov  5 10:00 /dev/ram3
brw----- 1 root kmem 1, 11 Nov  5 10:00 /dev/ram4
brw----- 1 root kmem 1, 12 Nov  5 10:00 /dev/ram5
```

У нас есть точки подключения для создания RAM-дисков, создаём диск:

```
# ramdisk 10000 /dev/ram5
size on /dev/ram5 set to 10000kB
```

Примечание: если вы попытаетесь узнать что-то детальнее по поводу `ramdisk`, то у вас мало что получится:

```
# man ramdisk
man: no manual on ramdisk
```

Разве что:

```
# ramdisk
usage: ramdisk <size in kB> [device]
```

Для более детального изучения придётся обратиться к исходному коду этой утилиты.

Диск мы создали, но теперь нужно создать на нём раздел (хотя бы один) и его форматировать:

```
# partition /dev/ram5 81:0+
```

Part	First	Last	Base	Size	Kb
0	0/001/00	8/063/31	32	18400	9200
1	0/000/00	0/000/-1	0	0	0
2	0/000/00	0/000/-1	0	0	0
3	0/000/00	0/000/-1	0	0	0

```
# fsck /dev/ram5
```

```
bad magic number in super block
fatal
```

```
# mkfs /dev/ram5
```

```
# fsck /dev/ram5
```

```
Checking zone map
```

```
Checking inode map
```

```
Checking inode list
```

```
blocksize = 4096          zonesize = 4096
```

```
  0   Regular files
  1   Directory
  0   Block special files
  0   Character special files
1279  Free inodes
  0   Named pipes
  0   Symbolic links
2475  Free zones
```

Теперь монтируем созданный диск в файловую систему MINIX3:

```
# mkdir /ram
# ls /
bin boot dev etc home lib mnt ram root sbin tmp usr var work
# mount /dev/ram5 /ram
/dev/ram5 is read-write mounted on /ram
```

Выполним испытательное копирование произвольного (достаточно большого) файла между подразделами диска MINIX3 и RAM-дискон:

```
# cd /home/olej
# ls -l file.*
-rw----- 1 root operator 1436656 Jan 20 13:51 file.tar.bz2
# time cp file.tar.bz2 /usr
 1.16 real  0.00 user  1.13 sys
# time cp file.tar.bz2 /ram
 1.15 real  0.00 user  0.96 sys
```

Различие во времени операции не столь значительно, но это, возможно, связано и с работой MINIX3 в среде виртуальной машины QEMU.

Источники информации

1. man на команду repartition

<http://minix1.woodhull.com/current/2.0.4/wwwman/man8/repartition.8.html>

<https://gforge.cs.vu.nl/gf/project/minix/scmsvn/?action=browse&path=%2Ftrunk%2Fsrc%2Fman%2Fman8%2Frepartition.8&view=log>

2. man на команду mkfs

<http://minix1.woodhull.com/current/2.0.4/wwwman/man1/mkfs.1.html>

<https://gforge.cs.vu.nl/gf/project/minix/scmsvn/?action=browse&path=%2Ftrunk%2Fsrc%2Fman%2Fman1%2Fmkfs.1&view=log>

3. man на команду fdisk

<http://minix1.woodhull.com/current/2.0.4/wwwman/man8/fdisk.8.html>

<https://gforge.cs.vu.nl/gf/project/minix/scmsvn/?action=browse&path=%2Ftrunk%2Fsrc%2Fman%2Fman8%2Ffdisk.8&view=log>

4. man на команду partition

<http://minix1.woodhull.com/current/2.0.4/wwwman/man8/partition.8.html>

<https://gforge.cs.vu.nl/gf/project/minix/scmsvn/?action=browse&path=%2Ftrunk%2Fsrc%2Fman%2Fman8%2Fpartition.8&view=log>

Примечания: в URL на man 1-я ссылка даётся на WWW текст для чтения, но относящийся к версии MINIX 2.04 (в большинстве совпадающий с MINIX 3), а 2-я — на SVN репозиторий проекта MINIX 3.