

## Лабораторное занятие 1. Знакомство с FreeBSD

### Задание 1. Установка ОС на виртуальную машину Virtual PC

1. Скопировать инсталляционные файлы ОС FreeBSD с ftp-сервера кафедры на локальную машину в папку d:\temp.

\\cs\ftp\distrib\OS\FreeBSD\6.1-RELEASE-i386-disc1.iso и  
\\cs\ftp\distrib\OS\FreeBSD\6.1-RELEASE-i386-disc2.iso.

2. Запустить управляющую программу (console) виртуальной машины Virtual PC. Сначала нужно создать новую машину (VM), запустив Мастер создания новой машины с помощью кнопки «New...». Отвечая на вопросы Мастера в последовательности окон, создаем VM с требуемыми значениями параметров.

а) «Create a virtual machine» - создаем новую VM;

б) имя машины можно оставить без изменения «New Virtual Machine», а ее расположением может быть выбрана папка d:\temp;

в) тип операционной системы остается «Other»;

г) объем используемой оперативной памяти можно оставить равным рекомендуемому в 128 Mb;

е) для размещения VM нужно выбрать новую область на жестком диске «A new virtual hard disk».

ж) размер области на жестком диске достаточен в 4096 Mb;

В папке d:\temp появятся два файла

«New Virtual Machine.vmc», «New Virtual Machine Hard Disk.vhd».

3. На созданную машину устанавливаем ОС FreeBSD 6.1, нажав на кнопку «Start». В появившемся окне «New Virtual Machine» с помощью меню «CD - Capture ISO Image...» виртуально устанавливаем CDROM «6.1-RELEASE-i386-disc1.iso». После этого начинается загрузка ОС с CDROM, которая позволит подготовить жесткий диск VM (!), перенести на него ОС и сделать жесткий диск загрузочным.

а) Первое диалоговое окно (ДО) позволяет выбрать язык общения с Мастером установки. Выбор «United States» нас вполне устраивает;

б) следующее ДО содержит главное меню установщика («Sysinstall Main Menu»). Нам надо выбрать пункт «Express - Custom», позволяющий пользователю задать или выбрать режимы программы установки;

в) в ДО «Choose Custom Installation Options», показанном на рис.1, нужно последовательно пройти все шаги от 2 до 7.

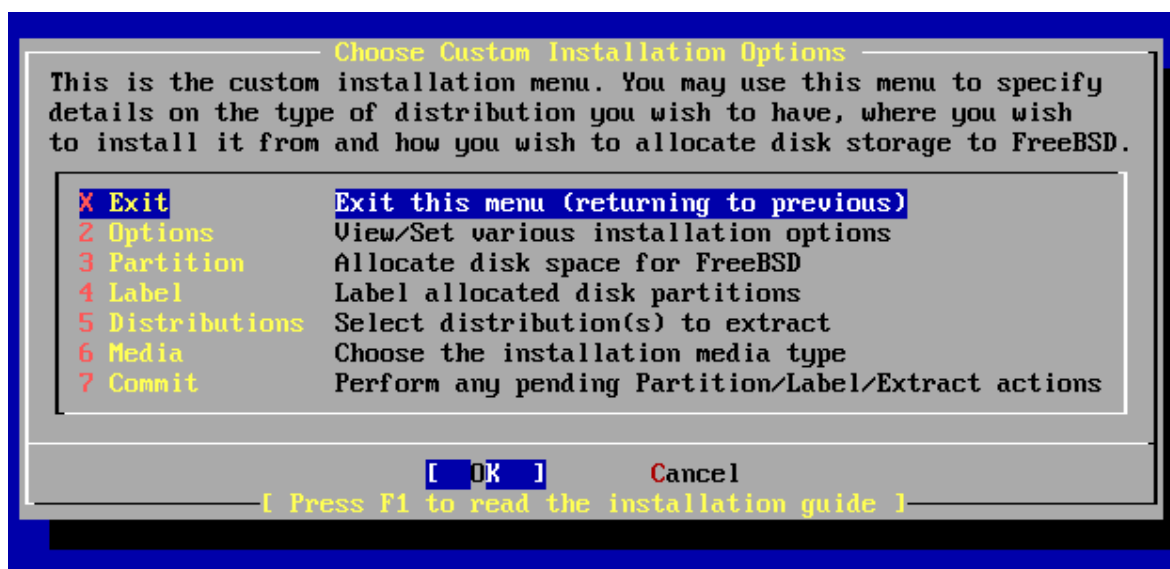


Рис. 1.

Пункт 2 позволяет просмотреть и установить режимы процесса установки. В первый раз можно просто открыть соответствующее ДО, ознакомиться с содержимым и вернуться назад, ничего не меняя. Для возврата на предшествующий уровень необходимо нажать клавишу 'Q', что указано в окне редактора режимов программы установки.

Пункт 3 необходим для выделения имеющегося свободного дискового пространства под устанавливаемую ОС FreeBSD – см. рис. 2.

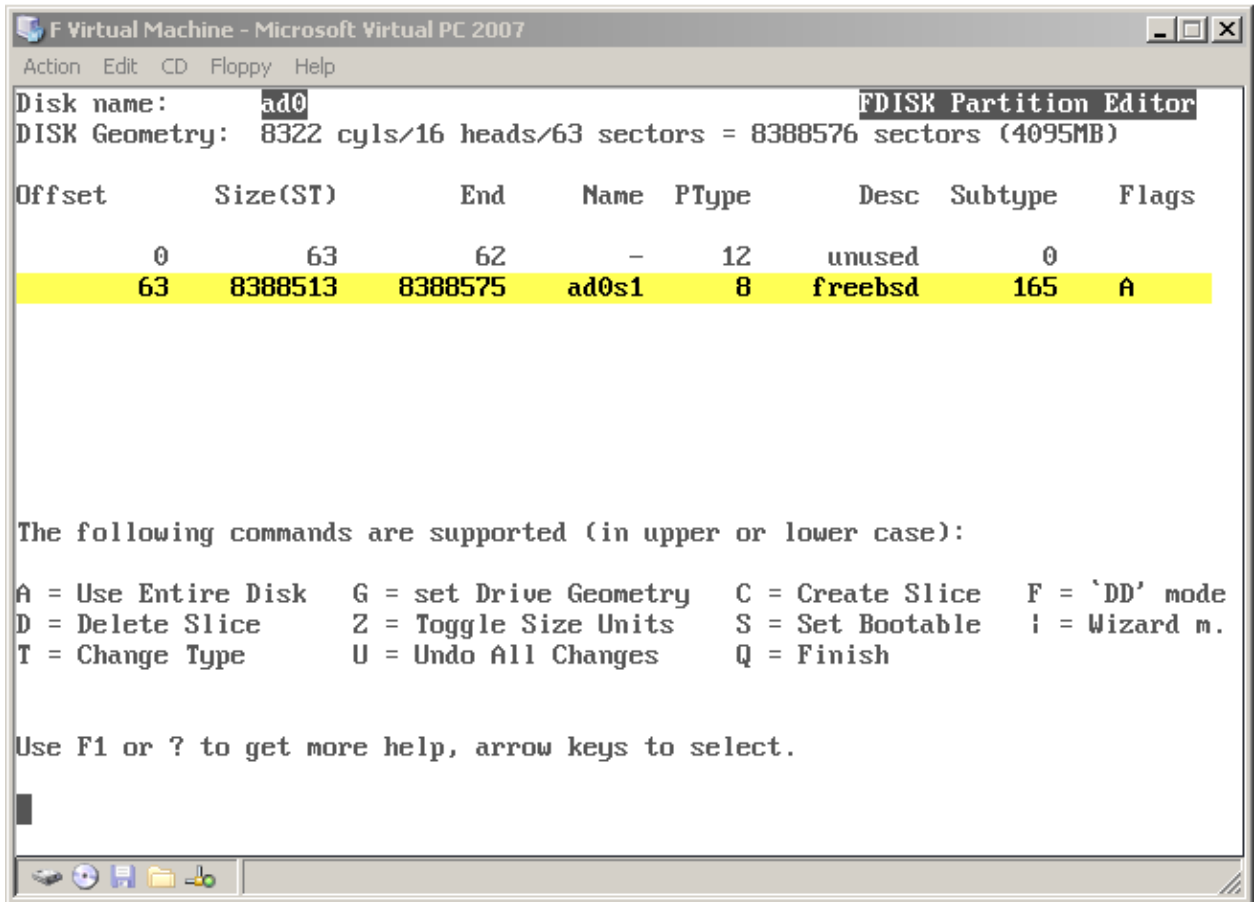


Рис. 2.

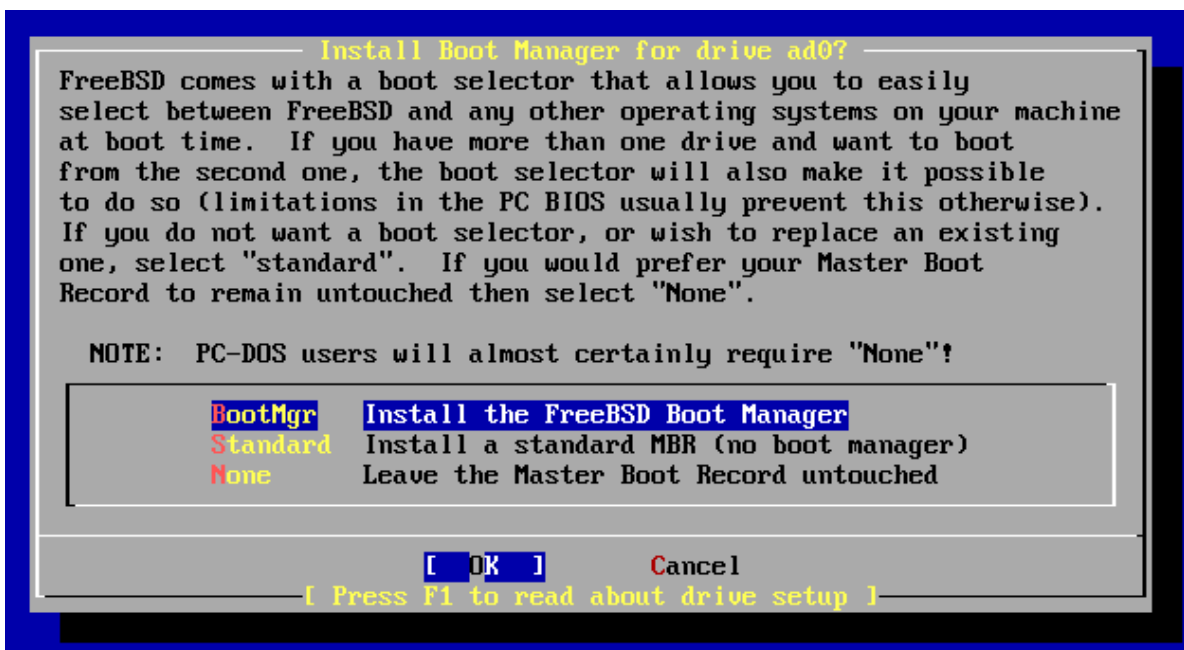


Рис. 3.

Для выделения всего пространства под ОС нужно нажать клавишу 'A'. Затем клавишей 'S' делаем диск загрузочным. Выход из ДО происходит по клавише 'Q'.

Попадаем в окно менеджера главной загрузочной записи – «Install Boot Manager», - показанное на рис. 3. Нужно выбрать пункт «Standard» - установка (сохранение) записи главного загрузчика.

Пункт 4 необходим для создания логических разделов внутри выделенного дискового пространства и присвоения им наименований, что показано на рис. 4.

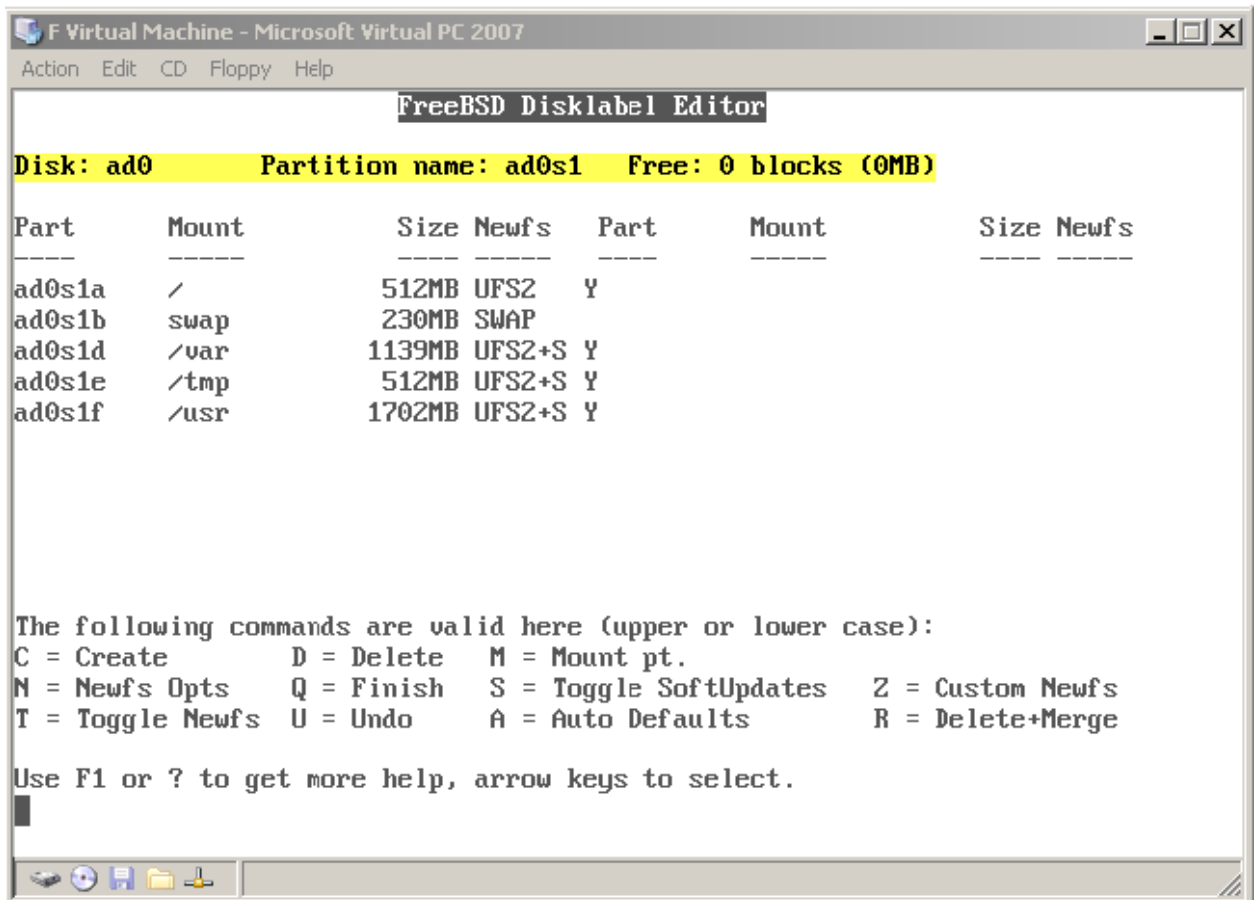


Рис. 4.

Минимально необходимыми являются разделы: корневой («/») и свопинга («**swap**»). Рекомендуется размер раздела под свопинг (выгрузку) устанавливать равным удвоенному размеру имеющейся оперативной памяти. Остальное место можно отдать корневому разделу. Для простоты можно воспользоваться стандартным распределением, нажав клавишу 'A'. Результат виден на рис. 4.

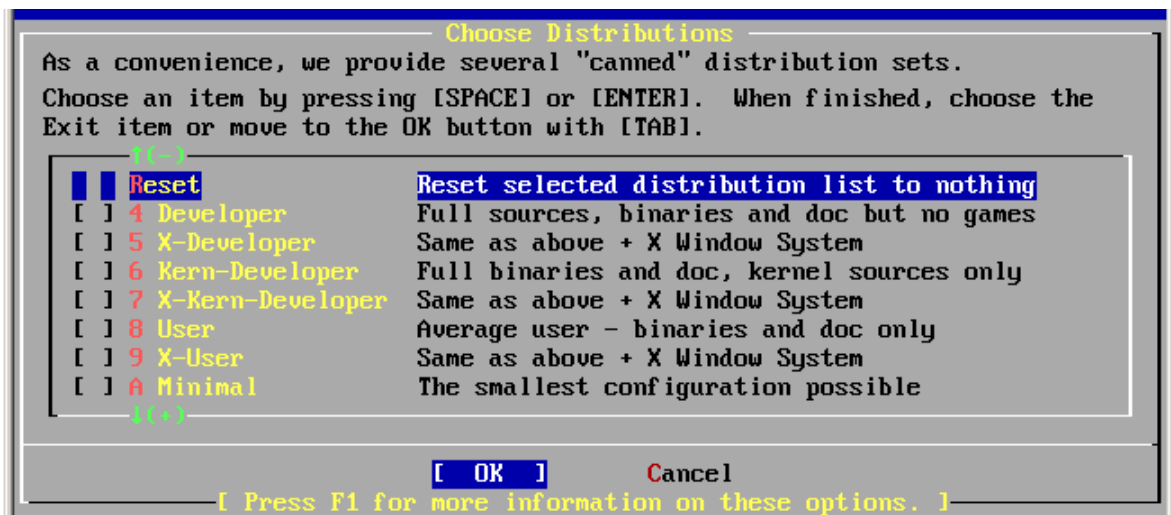


Рис. 5.

Пункт 5 («Distributions») позволяет выбрать состав устанавливаемого программного обеспечения – рис. 5. Начнем с варианта 'A' Minimal – минимально возможный работоспособный состав.

Пункт 6 («Media») необходим для выбора носителя с инсталляционным ПО – указываем пункт CD/DVD. Наконец пункт 7 («Commit») позволяет выполнить все незавершенные действия по установке ОС на жесткий диск. После выполнения установки и отказа от проверки выбранных режимов Мастер установки возвращает нас в ДО, показанное на рис. 1. Выбор пункта «Exit» переводит нас в ДО главного меню, и здесь мы должны нажать кнопку «Exit Install» (Завершение установки). Это приводит нас в окно подтверждения запроса – рис. 6.

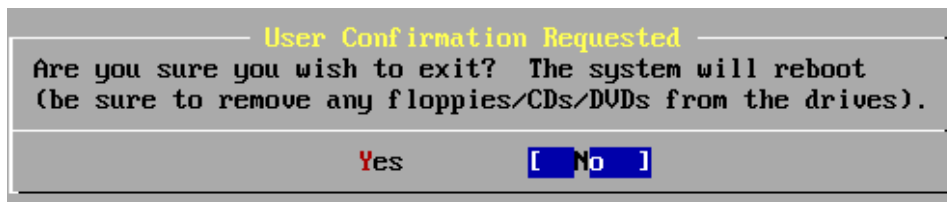


Рис. 6.

Прежде чем нажать на кнопку «Yes», через меню «CD - Release» VM виртуально вынимаем CD с инсталляционным ПО из устройства CD/DVD. После нажатия на «Yes» и небольшой паузы начинается процесс перезагрузки ОС уже с подготовленного жесткого диска. Загрузка завершается приглашением «login:» - нужно ввести имя пользователя. Единственным пользователем на данный момент является главный пользователь **root**. После аутентификации пользователя root на экране появляется приглашение оболочки – интерпретатора команд, - в виде символа '#'. Это приглашение именно для пользователя root. Система готова к работе.

Чтобы выйти из системы (и получить новое приглашение login:) наберите

```
# exit
```

столько раз, сколько нужно. Да, нажимайте enter после набора команд, и помните, что UNIX чувствителен к регистру букв--набирайте exit, но не EXIT.

Для завершения работы машины наберите

```
# shutdown -h now
```

После остановки работы ОС для закрытия VM нужно выбрать пункты меню «Action - Close», а затем в появившемся окне диалога выполнить выключение VM – Turn Off.

Для перезагрузки нужно набрать

```
# shutdown -r now
```

или

```
# reboot
```

## Задание 2. Установка дополнительных компонентов FreeBSD

Операционные системы семейства Unix отличаются от альтернативных тем что идут в комплекте с большим количеством полезных и удобных программ. В FreeBSD эти программы называются packages – пакеты, - уже скомпилированные и ports – порты, - доступные только в исходных текстах.

Для установки дополнительных компонентов необходимо запустить программу

```
# sysinstall
```

и перед нами снова откроется ДО главного меню установщика – см рис. 7.

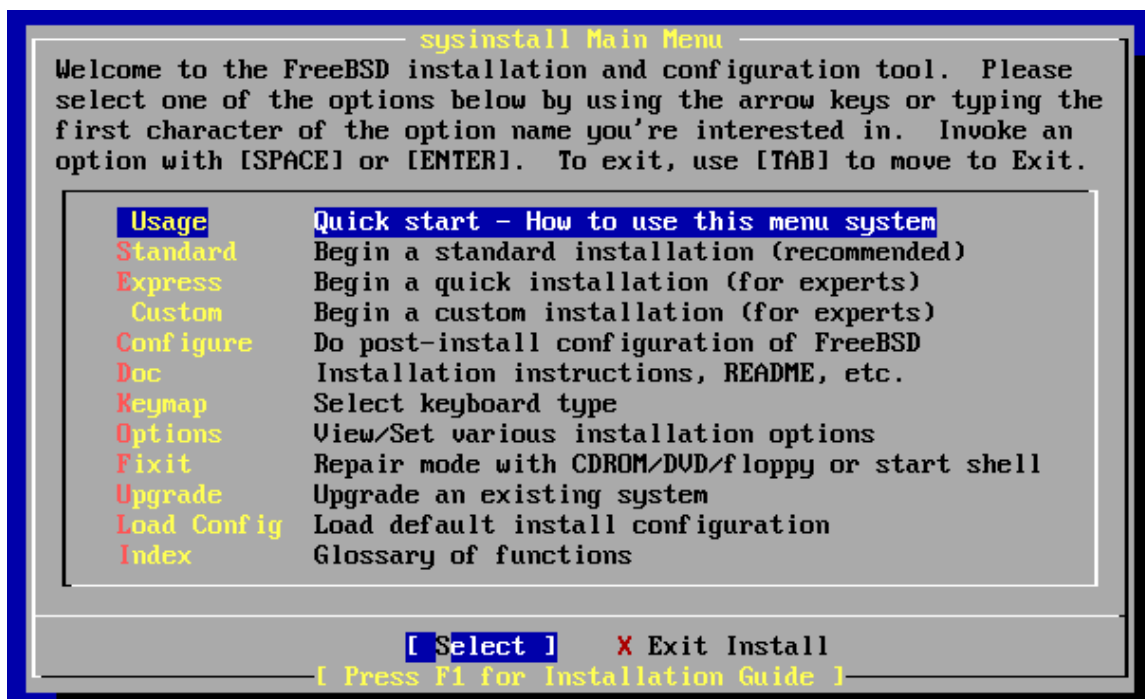


Рис. 7.

Выбрав пункт Configure, мы через ДО меню конфигурации, показанное на рис. 8, получаем возможность установить дополнительные дистрибутивные наборы (Distributions) и пакеты (Packages).

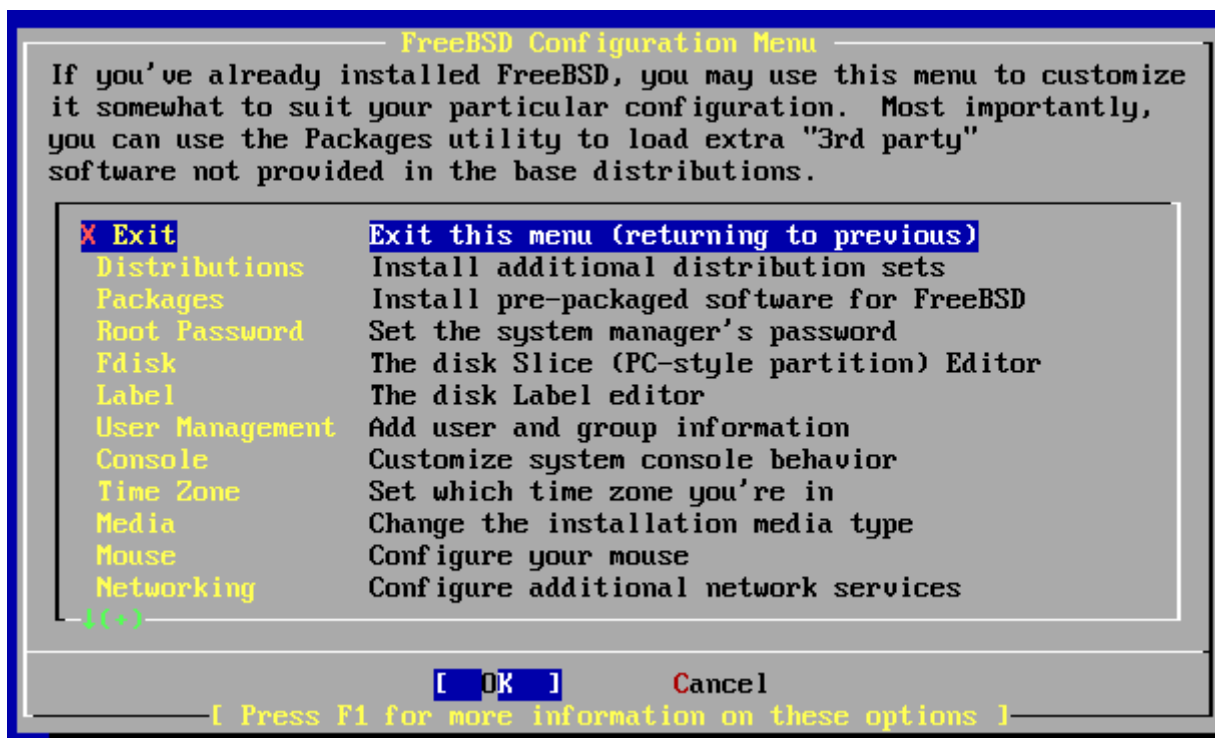


Рис. 8.

Из дистрибутивных наборов выберем для установки doc (разнообразная документация по ОС), man (страницы документации, старый стиль), info (страницы документации,

современный GNU стиль), `src/sys` (исходные тексты ядра), `src/base` (исходные тексты базовой системы). Здесь нужно не забыть через меню ВМ «CD - Capture ISO Image . . .» виртуально поставить носитель `6.1-RELEASE-i386-disc1.iso`.

Из набора пакетов рекомендуется поставить `shells/bash` (удобный командный интерпретатор), `editors/emacs`, `editors/vim` (популярные текстовые редакторы и среды разработки), `devel/gmake` (GNU make, наиболее распространен, даже Microsoft `nmake` совместим с ним). При установке пакетов Мастер требует смены носителей: с `6.1-RELEASE-i386-disc1.iso` на `6.1-RELEASE-i386-disc2.iso` и наоборот. Такая замена может производиться с помощью меню ВМ «CD - Capture ISO Image . . .».

Если при установке некоторых компонентов произошли сбои, процедура может быть повторена с помощью программы `sysinstall`.

### Задание 3. Добавление пользователя с привилегиями `root`

Если при установке системы вы не создали ни одного пользователя, и поэтому вошли в систему как `root`, то теперь вы должны создать пользователя по команде

```
# adduser
```

При первом использовании утилиты `adduser` она может запрашивать сохранение некоторых параметров для использования их по умолчанию. Вы можете сделать оболочкой, используемой по умолчанию, командный процессор `bash`, а не `sh`, если по умолчанию вам предлагается `sh`. В противном случае просто нажимайте `enter` для принятия всех предлагаемых по умолчанию вариантов. Эти значения по умолчанию сохраняются в файле `/etc/adduser.conf`, в форме, доступной для редактирования.

Предположим, что вы создали пользователя `jack` с полным именем *Jack Benimble*. Назначьте пользователю `jack` пароль, если информационная безопасность имеет значение (даже если это дети, которые могут стучать по клавиатуре). Когда вам будет задан вопрос по включению пользователя `jack` в другие группы, наберите `wheel`

```
Login group is 'jack'. Invite jack into other groups: wheel
```

Это позволит входить в систему как пользователь `jack` и использовать команду `su` для того, чтобы стать пользователем `root`. Тогда вас не будут больше обвинять в том, что вы входите в систему как пользователь `root`.

Вы можете прекратить работу с `adduser` в любой момент, нажав **Ctrl-C**, а в завершении ввода у вас будет шанс подтвердить заведение нового пользователя или набрать **n** в качестве отрицательного ответа. Вам может захотеться создать второго нового пользователя, для того, чтобы при редактировании файлов для входа пользователя `jack` имелся горячий резерв на тот случай, если что-то пойдёт не так.

После того, как вы это сделаете, воспользуйтесь командой `exit` для возврата к приглашению ко входу в систему и зарегистрируйтесь в ней как пользователь `jack`. Вообще говоря, лучше всего основную массу работы выполнять, работая как обычный пользователь, который не имеет мощь и опасность пользователя `root`.

Если вы уже создали пользователя и хотите, чтобы он мог выполнять команду `su` для получения привилегий `root`, вы можете войти в систему как `root` и отредактировать файл `/etc/group`, добавив пользователя `jack` в первую строчку (в группу `wheel`). Однако сначала вам нужно поупражняться с программой `vim`, текстовым редактором.

Для удаления пользователя воспользуйтесь командой `rmuser`.

## Задание 4. Написание программ в среде FreeBSD

Основные команды работы с текстовым редактором `vim` описаны в приложении 1. Компиляция и запуск программ производятся следующим образом. Предположим, что все файлы создаются в домашнем каталоге. Пусть необходимо скомпилировать программу из исходника `test.c` с помощью компилятора `gcc`. Сначала получим объектный файл:

```
gcc -c test.c
```

Если теперь посмотреть содержимое домашнего каталога, то увидим новый файл `t.o`, что обеспечено опцией `-c`. Получение загрузочного файла из объектного производится программой компоновщиком. Программа `gcc` может выполнить и работу компоновщика:

```
gcc -o test test.o
```

Созданный загрузочный файл будет называться `test` (без расширения). Опция `-o` позволяет в командной строке задать имя загрузочного файла. Запуск программы из рабочей папки производится командой:

```
./test
```

Теперь опишем собственно задание.

1. В домашнем каталоге написать программу с помощью редактора `vim`:

```
#include <stdio.h>

int main(void)
{
    printf("Hello world!\n");
    return 0;
}
```

2. Скомпилировать и запустить программу.

## Задание 5. Работа с утилитой make

`Make` – это программа, которая обычно используется при разработке больших программ, состоящих из нескольких файлов. Она следит за тем, какие объектные файлы зависят от каких исходных файлов и файлов заголовков. При вызове она выполняет минимальное количество перекомпиляций для обновления целевого файла.

Зависимости файлов задаются в файлах с именами `makefile` или `Makefile`, либо определяются ключом `-f`. `Make` имеет несколько встроенных правил, например, она знает, как получить файлы `*.o` из файлов `*.c`.

Пример файла `makefile`:

```
prog: head.o tail.o                # цель prog зависит от файлов head.o и tail.o
    gcc -o prog head.o tail.o      # сообщает make, как достичь цели program
    @echo Program done.           # извещает о завершении процесса компиляции
head.o: head.c                    # цель head.o зависит от файла head.c
    gcc -c head.c
tail.o: tail.c                    # цель tail.o зависит от файла tail.c
    gcc -c tail.c
```

Обратите внимание, что при указании правила достижения цели (в данном примере это вызовы `gcc` и `echo`) используются символы табуляции (без пробелов).

Компиляция с помощью `make` требует предварительного создания файла `makefile` в рабочей папке в формате, аналогичном показанному выше примеру.

Вызов утилиты `make`:

```
make
```

Для создания файла `makefile` надо использовать редактор `emacs`, основные команды которого описаны в приложении 1.

Собственно задание заключается в создании `makefile`'а для компиляции и компоновки вашей программы на языке C, например `test.c`. Затем утилита `make` на основе `makefile`'а строит загрузочную программу. Эта программа при запуске должна правильно работать.