

1 лаба ОС ReactOS (1 стр) и netBSD (2 стр)
ReactOS для OS X (mac OS)

1. Сначала нужно установить RosBE. Скачиваем отсюда исходники:

https://www.reactos.org/wiki/Build_Environment

Затем в консоли пишем ./RosBE-Builder.sh

чтобы запустить:

```
cd /usr/local/RosBE/
```

```
./RosBE.sh
```

2. <https://tortoisesvn.net> на мак не предусмотрено, поэтому скачиваем аналог.

```
brew install svn
```

Далее переходим в папку, в которой будет рабочая директория и в терминале пишем следующее: `svn checkout https://svn.reactos.org/reactos/trunk/`

3. Основной debug.log файл внутри реактоса можно увидеть, если зайти в операционку с параметром Log File, далее зайти в локальный диск C и там в папку реактоса. Нам нужно отредактировать файл из ядра. Самый простой способ: посмотреть файл дебаг лога и выбрать оттуда любой из папки ntoskrnl. Допустим, driver.c

4. Запускаем RosBE, `cd PATH_TO_SRC/reactos`

```
./configure.sh
```

Затем переходим в следующую папку: `PATH_TO_SRC/reactos/output-MinGW-i386`

в консоль: `ninja bootcd`

ISO будет лежать в папке указанной выше

5. Устанавливаем в виртуал боксе реактось с собранного ISO-образа. По установке проблем возникнуть никаких не должно, почти везде на автомате можно нажимать enter

netBSD для OS X (mac OS)

1. Устанавливаем GITHUB

export PKG_PATH=<ftp.netbsd.org/pub/pkgsrc/packages/NetBSD/i386/8.0/All>
pkg_add -v git

2. Качаем исходник NetBSD с GITHUB

git clone --depth=1 --branch netbsd-8 <https://github.com/NetBSD/src.git>

3. В файле /usr/src/sys/kern/init_main.c добавляем своё ФИО.

4. Из каталога /usr/src пишем: ./build.sh -U -u -j4 -m amd64 -N 0 -O ~/obj release iso-image

Как собрать ядро?

1. Проверьте, что Вы установили набор компиляторов comp.tgz, шедший с основной системой
2. Скачайте и распакуйте исходники ядра
3. Если Вы скачали -current, то возможно должны скомпилировать новый набор инструментов. Если Вы полностью скачали исходники: `cd /usr/src && ./build.sh tools`
4. `"cd /sys/arch/ARCH/conf"`
ARCH - Ваша архитектура, например 'i386', 'sparc', 'mac68k'.
5. `"cp GENERIC MYCONF"`
MYCONF имя вашей новой конфигурации. Вы можете использовать имя хоста, тип машины или Ваше имя, по желанию. Допускаются цифры и буквы латинского алфавита и `_,-`.
6. Редактирование MYCONF.
Первоначально Вы можете пропустить эту стадию. Вы можете удалить драйверы для типов центрального процессора, аппаратных средств, и устройств, типа которых Вы не имеете или не используете. Также для i386 можно закоментировать линию 'pc0' и открыть линию 'vt0' для появления виртуальных консолей. Хорошим началом будет изучить вывод dmesg. Прочитайте "options(4)" для получения информации по отдельным опциям ядра.
7. `"config MYCONF"`
генерирует каталог сборки ядра.
8. `"cd ../compile/MYCONF"`
перейдем в этот каталог
9. `"make depend"`
сгенерирует нам файл зависимостей '.depend'
10. `"make"`
откомпилирует ядро. В конце Вы получите 'netbsd' ядро. Этот процесс займет много времени на VAX, мало на большой Alpha и все остальное в промежутке.
11. `"mv /netbsd /netbsd.old ; mv /sys/arch/ARCH/compile/MYCONF/netbsd /"`
это сохранит Ваше старое ядро (это ОЧЕНЬ важно) и ставит новое ядро.
12. `"reboot"`
перезагружает компьютер, используя при следующей загрузке уже новое ядро. Выводит при загрузке сообщение типа: 'NetBSD VERSION (MYCONF) #0: COMPILER_DATE'
13. Если у Вас проблемы:
Вы должны загрузить 'netbsd.old' ядро в режиме single user. Для выполнения этой процедуры на i386 необходимо выполнить:
Нажать SPACE, когда появится первое сообщение NetBSD
Выполнить `boot netbsd.old -s`
Возвращайте старое ядро:
`fsck /`
`mount /`
`mv netbsd.old netbsd`
`exit`

5. ISO-image будет создан в каталоге /root/obj/releasedir/images

далее все как и с реактосью, за исключением того, что при установке придется менять некоторые параметры под свой компьютер.