

# Оглавление

<b>1</b>	<b>Цель работы</b>	<b>2</b>
<b>2</b>	<b>Сборка дистрибутива ReactOS</b>	<b>3</b>
2.1	Постановка задачи . . . . .	3
2.2	Порядок выполнения работы . . . . .	3
2.3	Практическая реализация . . . . .	4
2.4	Результаты . . . . .	5
<b>3</b>	<b>Операционная система NetBSD: среда сборки, установка и тестирование</b>	<b>6</b>
3.1	Постановка задачи . . . . .	6
3.2	Порядок выполнения работы . . . . .	6
3.3	Практическая реализация . . . . .	7
3.4	Результаты . . . . .	8
<b>4</b>	<b>Выводы</b>	<b>10</b>

## 1 Цель работы

Целью данной лабораторной работы является приобретение навыков по самостоятельной сборке «с нуля» дистрибутивов операционных систем ReactOS и NetBSD с последующей их установкой в виртуальную машину.

## 2 Сборка дистрибутива ReactOS

### 2.1 Постановка задачи

Необходимо настроить среду сборки и тестирования ReactOS на своём компьютере (или на компьютере в учебной аудитории университета), собрать установочный образ и произвести его установку в виртуальную машину с помощью программных средств виртуализации.

### 2.2 Порядок выполнения работы

Для выполнения лабораторной работы необходимо реализовать следующие этапы:

1. Получить ReactOS Build Environment (RosBE) по ссылке со страницы [https://reactos.org/wiki/Building\\_ReactOS](https://reactos.org/wiki/Building_ReactOS) и установить её согласно прилагаемому к ней руководству.
2. Установить систему управления версиями Git (например, TortoiseGit + MSysGit для Windows).
3. Создать у себя рабочую копию (working copy) исходных кодов основной ветки разработки `master` из репозитория <https://github.com/reactos/reactos.git>.
4. Произвести сборку установочного ISO-образа операционной системы ReactOS.
5. Установить полученный образ в виртуальной машине (Virtual Box), настроив вывод отладочной информации (debug log) в файл.
6. Модифицировать ядро операционной системы (ntoskrnl) так, чтобы в отладочный лог в момент инициализации ядра выводилась фамилия студента, выполнившего лабораторную работу.
7. Заменить ядро в установленной ОС в виртуальной машине на модифицированное и загрузить снова, убедившись в наличии нужной информации в debug log.

## 2.3 Практическая реализация

1. Скачаем и установим необходимые для начала работы компоненты:

- VirtualBox 6.0.24 и VirtualBox Extension Pack
- Система контроля версий Git
- ReactOS Build Environment (далее – RosBE)

2. Запустим RosBE и выполним в ней следующие команды:

```
1 | git clone https://github.com/reactos/reactos
2 | cd reactos
3 | configure.cmd
4 | cd output-MinGW-i386
5 | ninja bootcd
```

После выполнения данных команд получим установочный ISO-образ ReactOS, который будет иметь адрес `.../reactos/output-MinGW-i386/bootcd.iso`.

3. Установим и запустим ReactOS в VirtualBox.

4. При помощи VirtualBox Extension Pack и общих папок наладим обмен файлами между домашней операционной системой (далее – ДОС) и ReactOS.

5. Настроим вывод отладочной информации ReactOS через COM-порт VirtualBox в файл, который находится в ДОС.

6. Модифицируем файл `.../reactos/ntoskrnl/kd64/kdinit.c` следующим образом:

```
1 | ...
2 | static CODE_SEG("INIT")
3 | VOID
4 | KdpPrintBanner(IN SIZE_T MemSizeMBs)
5 | {
6 |     DPRINT1("-----\n");
7 |     DPRINT1("ReactOS " KERNEL_VERSION_STR " (Build "
              KERNEL_VERSION_BUILD_STR ") (Commit "
              KERNEL_VERSION_COMMIT_HASH ")\n");
```

```

8      DPRINT1("BMSTU IU9 OS lab1 IU9-42B Chernikov Artem\n");
9      DPRINT1("%u System Processor [%u MB Memory]\n",
              KeNumberProcessors, MemSizeMBs);
10
11     if (KeLoaderBlock)
12     {
13         DPRINT1("Command Line: %s\n",
                  KeLoaderBlock->LoadOptions);
14         DPRINT1("ARC Paths: %s %s %s %s\n",
                  KeLoaderBlock->ArcBootDeviceName,
                  KeLoaderBlock->NtHalPathName,
                  KeLoaderBlock->ArcHalDeviceName,
                  KeLoaderBlock->NtBootPathName);
15     }
16 }
17 ...

```

7. В ранее открытой RosBE выполним следующие команды:

```
1 | ninja ntoskrnl
```

После выполнения данных команд получим модифицированное ядро ReactOS, которое будет иметь адрес `.../reactos/output-MinGW-i386/ntoskrnl.exe`.

8. Заменяем оригинальное ядро ReactOS, которое имеет адрес `C:/ReactOS/system32/ntoskrnl.exe`, на модифицированное ядро.

9. Перезапустим ReactOS в режиме Debug.

## 2.4 Результаты

```

(..boot/freeldr/freeldr/disk/partition.c:425) fixme: DiskGetPartitionEntry() unimplemented for RAW
(..boot/freeldr/freeldr/arch/i386/hwdisk.c:133) err: SectorSize (2048) != Geometry.BytesPerSector (0), expect problems!
(ntoskrnl/kd64/kdinit.c:74) -----
(ntoskrnl/kd64/kdinit.c:75)
(ntoskrnl/kd64/kdinit.c:76) By Chernikov Artem, IU9-42B
(ntoskrnl/kd64/kdinit.c:77)
(ntoskrnl/kd64/kdinit.c:78) -----
(ntoskrnl/kd64/kdinit.c:79) ReactOS LAB0.4.15-x86-dev (Build 20240225-0.4.15-dev-7674-gc0b4db1) (Commit
c0b4db14de2af1651cf67a067346c41842f62e4f)

```

Рисунок 2.1 – Скриншот журнала отладки ReactOS

## 3 Операционная система NetBSD: среда сборки, установка и тестирование

### 3.1 Постановка задачи

Необходимо:

1. Установить новый выпуск NetBSD из дистрибутива с официального сайта проекта NetBSD <https://www.netbsd.org/> в виртуальную машину.
2. Настроить среду сборки и тестирования NetBSD в виртуальной машине на своём компьютере (или на компьютере в учебной аудитории университета).
3. Пересобрать ядро в этой виртуальной машине, добавив вывод фамилии обучающегося в debug log.

**NB!** Не следует предпринимать попытки кросс-компилировать ядро NetBSD в Windows, Linux или OS X. Собирать его нужно внутри самой NetBSD, которая установлена в виртуальной машине.

### 3.2 Порядок выполнения работы

Для выполнения лабораторной работы необходимо реализовать следующие этапы:

1. Скачать с сайта <https://netbsd.org/> образ дистрибутива ОС NetBSD и установить его в виртуальную машину (VirtualBox), настроив вывод отладочной информации (debug log) в файл.
2. Настроить среду сборки и скачать исходный код системы.
3. Модифицировать ядро операционной системы так, чтобы в отладочный лог в момент инициализации ядра выводилась фамилия студента, выполнившего лабораторную работу.
4. Заменить ядро в установленной ОС в виртуальной машине на модифицированное и загрузить снова, убедиться в наличии нужной информации в debug log.

### 3.3 Практическая реализация

1. Скачаем установочный ISO-образ NetBSD 9.4 и установим данную операционную систему в VirtualBox. Стоит отметить, что для удобства эксплуатации NetBSD дополнительно было установлено и настроено:
  - Соединение с сетью Интернет через сетевой мост VirtualBox.
  - Пакетный менеджер pkgin.
  - Текстовый редактор nano.
  - SSH-сервер с доступом от имени пользователя **root**.
2. Запустим NetBSD. После запуска операционной системы и входа в неё под именем пользователя **root** мы будем находиться в директории **/root**.
3. Скачаем и распакуем архив с исходным кодом NetBSD в корневую директорию **/** с помощью следующих команд:

```
1 ftp -i
   ftp://ftp.de.netbsd.org/netbsd/NetBSD-9.4/source/sets/
2 mget syssrc.tgz
3 quit
4
5 tar -xzf syssrc.tgz -C /
```

4. Модифицируем файл **/usr/src/sys/kern/init\_main.c** следующим образом:

```
1 ...
2 static CODE_SEG("INIT")
3 VOID
4 KdpPrintBanner(IN SIZE_T MemSizeMBs)
5 {
6     DPRINT1("-----");
7     DPRINT1("ReactOS " KERNEL_VERSION_STR " (Build "
8             KERNEL_VERSION_BUILD_STR ") (Commit "
9             KERNEL_VERSION_COMMIT_HASH ")\n");
10    DPRINT1("BMSTU IU9 OS lab1 IU9-42B Chernikov Artem\n");
11    DPRINT1("%u System Processor [%u MB Memory]\n",
12            KeNumberProcessors, MemSizeMBs);
13
14    if (KeLoaderBlock)
```

```

12     {
13         DPRINTF("Command Line: %s\n",
14                 KeLoaderBlock->LoadOptions);
15         DPRINTF("ARC Paths: %s %s %s %s\n",
16                 KeLoaderBlock->ArcBootDeviceName,
17                 KeLoaderBlock->NtHalPathName,
18                 KeLoaderBlock->ArcHalDeviceName,
19                 KeLoaderBlock->NtBootPathName);
20     }
21 }
22 ...

```

5. Создадим конфигурацию модифицированного ядра и подготовим файлы для его сборки в соответствии с этой конфигурацией при помощи следующих команд:

```

1 cd /usr/src/sys/arch/amd64/conf/
2 cp GENERIC BMSTU_IU9_OS
3 config BMSTU_IU9_OS

```

6. Соберём зависимости и модифицированное ядро с помощью следующих команд:

```

1 cd ../compile/BMSTU_IU9_OS
2 make depend
3 make

```

7. Заменяем оригинальное ядро на модифицированное при помощи следующих команд:

```

1 mv /netbsd /netbsd.old
2 mv netbsd /

```

8. Перезапустим NetBSD.

## 3.4 Результаты



```

[ 4.849793] wd0: GPT GUID: bade135c-4ca3-42c3-9220-e1ee75aa1418
[ 4.849793] dk0 at wd0: "655ca137-f800-4d97-b258-2a923ecda6f8", 27097021 blocks at 64, type: ffs
[ 4.849793] dk1 at wd0: "fefe16a4-3b7c-4094-9024-0085b3a02ac9", 6457311 blocks at 27097088, type: swap
[ 4.863996] wd0: 32-bit data port
[ 4.863996] wd0: drive supports PIO mode 4, DMA mode 2, Ultra-DMA mode 6 (Ultra/133)
[ 4.863996] wd0(piixide0:0:0): using PIO mode 4, Ultra-DMA mode 2 (Ultra/33) (using DMA)
[ 4.900092] WARNING: 1 error while detecting hardware; check system log.
[ 4.900092] boot device: wd0
[ 4.910217] root on dk0 dumps on dk1
[ 4.920293] root file system type: ffs
[ 4.930374] kern.module.path=/stand/amd64/9.3/modules
[ 15.865256] wsdisplay0: screen 1 added (80x25, vt100 emulation)
[ 15.873634] wsdisplay0: screen 2 added (80x25, vt100 emulation)
[ 15.873634] wsdisplay0: screen 3 added (80x25, vt100 emulation)
[ 15.873634] wsdisplay0: screen 4 added (80x25, vt100 emulation)
[ 3530.042710] Driver LAB2 init by Artem Chernikov
[ 3576.306657] WARNING: module error: module with name `lab2' already loaded
[ 3655.300170] WARNING: module error: module with name `lab2' already loaded
[ 3795.171271] Driver LAB2 unloaded
[ 3800.184717] Driver LAB2 init by Artem Chernikov
localhost# █

```

Рисунок 3.1 – Скриншот журнала отладки NetBSD

## 4 Выводы

В рамках данной лабораторной работы были получены базовые навыки сборки NT- и UNIX-подобных операционных систем из их исходных кодов, а также модификации их ядер.

## Список литературы

- [1] VirtualBox // ReactOS Wiki. URL: <https://reactos.org/wiki/VirtualBox> (дата обращения: 20.05.2024).
- [2] Building ReactOS // ReactOS Wiki. URL: [https://reactos.org/wiki/Building\\_ReactOS](https://reactos.org/wiki/Building_ReactOS) (дата обращения: 20.05.2024).
- [3] Kernel // NetBSD Documentation. URL: <https://netbsd.org/docs/guide/en/chap-kernel.html> (дата обращения: 20.05.2024).