

[Главная » СПО](#)

СПО. Лабораторная работа №7

Создание простейшей ОС

[▶ Оглавление](#)

Задание

1. Установить и настроить необходимые для сборки ОС инструменты.
2. Проанализировать и доработать исходный код.
3. Откомпилировать и запустить ОС.

Ход работы

1. Подготовка и настройка среды

Установим необходимые пакеты.

Debian-based

```
sudo apt install -y build-essential bison flex libgmp3-dev libmpc-dev libr
```

Arch-based

```
sudo pacman -Sy base-devel gmp libmpc mpfr xorriso
```

Настроим переменные среды:



```
export PREFIX="$HOME/opt/cross"  
export TARGET=i686-elf  
export PATH="$PREFIX/bin:$PATH"
```

2. Сборка кросс-компилятора

Создадим и перейдём в директорию, где будет осуществляться сборка:

```
mkdir ~/src && cd ~/src
```

Скачаем, разархивируем, соберём и установим binutils:

```
wget https://ftp.gnu.org/gnu/binutils/binutils-2.40.tar.xz  
### Если не установлен wget, можно вместо этой команды выполнить:  
curl https://ftp.gnu.org/gnu/binutils/binutils-2.40.tar.xz -o binutils-2.4  
  
tar -xf binutils-2.40.tar.xz  
rm binutils-2.40.tar.xz  
mkdir build-binutils && cd build-binutils  
../binutils-2.40/configure --target=$TARGET --prefix="$PREFIX" --with-sysr  
make  
make install
```

Скачаем, разархивируем, соберём и установим кросс-компилятор gcc:

```
cd ~/src  
wget https://ftp.gnu.org/gnu/gcc/gcc-12.2.0/gcc-12.2.0.tar.xz  
tar -xf gcc-12.2.0.tar.xz  
rm gcc-12.2.0.tar.xz  
mkdir build-gcc && cd build-gcc  
../gcc-12.2.0/configure --target=$TARGET --prefix="$PREFIX" --disable-nls  
make all-gcc  
make all-target-libgcc
```

```
make install-gcc  
make install-target-libgcc
```

Рекомендую запастись терпением и чаем с печеньками, т.к. сборка компилятора - это довольно долго (на современном железе под ВМ уйдёт ~5-10 мин).

3. Сборка ОС

```
export PATH=$HOME/opt/cross/bin:$PATH  
  
mkdir ~/src/myos && cd ~/src/myos  
wget https://evil-teacher.on.fleek.co/ssw/lab07_src.zip  
unzip lab07_src.zip  
make
```

4. Запуск ОС

QEMU

Установка QEMU:

Debian-based

```
sudo apt install -y qemu qemu-kvm
```

Arch-based

```
sudo pacman -Sy qemu-desktop
```

Запуск ОС:

```
qemu-system-i386 -cdrom myos.iso
```

VirtualBox



Создайте виртуальную машину с типом Other и версией Other/Unknown. Объём оперативной памяти можно оставить 64 Mb, жёсткий диск создавать не требуется. В порядке загрузки оставляем только оптический диск и в носителях под контроллером IDE монтируем `myos.iso`, собранный в предыдущем шаге. Остальные настройки оставляем по умолчанию. Если всё выполнено верно, при запуске виртуальной машины появится меню загрузчика GRUB, через которое система будет загружена.

Также вы можете записать файл `myos.iso` на флешку и запустить эту ОС на физической машине, если ваш процессор поддерживает выполнение в режиме совместимости с архитектурой i386-i686 (большинство современных x86 процессоров так ещё поддерживают).

Поздравляю, вы только что собрали и запустили свою первую ОС!

Почитать

Для интересующихся оставляю полезные ресурсы:

1. [OS Dev Wiki](#)
2. [Operating Systems: From 0 to 1](#)
3. [Writing a Simple Operating System — from Scratch by Nick Blundell](#)
4. [os-tutorial - How to create an OS from scratch!](#)
5. [Linux From Scratch](#)
6. [Erik Helin, Adam Renberg - The little book about OS development](#)
7. [Operating System Development Series](#)
8. Статьи Арджуна Сридхарана: [kernel 101](#) и [kernel 201](#)

← ПРЕДЫДУЩАЯ

СПО. Лабораторная работа №6

