

# Compiling KOS on Linux

## Contents

- 1 Overview
- 2 Install script
- 3 Preparations
- 4 Downloading KOS
- 5 Toolchain (cross-compiler and libraries)
- 6 Setting up KOS
- 7 KOS-Ports

## Overview

This tutorial is a step-by-step guide on how to setup a toolchain and KOS environment on your GNU/Linux system.

The toolchain consists of a C/C++ compiler (GCC), assembler and linker (binutils), and C library (newlib). As the Dreamcast has two processors - the SH4 CPU and the AICA (ARM) sound processor - the toolchain includes compilers for both.

KOS consists of the operating system core (kos) and a set of nicely integrated libraries (kos-ports).

## Install script

Please consider trying this install script first: File:Kos setup script.zip. It will perform the steps below automatically.

## Preparations

You need the following software installed:

- git
- make (build-essential package)
- tar/gzip/bzip2
- gcc/g++
- development packages of libjpeg and libpng
- patch
- texinfo (for makeinfo etc)
- wget

## Downloading KOS

KOS is available through a Git repository at SourceForge. The standard install directory assumed in the configuration files is `/opt/toolchains/dc/{kos, kos-ports}`.

```
$ git clone git://git.code.sf.net/p/cadcdev/kallistios /opt/toolchains/dc/kos
```

## Toolchain (cross-compiler and libraries)

After cloning the KOS repository, run the toolchain download+unpack+compile scripts:

```
$ cd /opt/toolchains/dc/kos/utils/dc-chain  
$ sh download.sh  
$ sh unpack.sh
```

For compilation of the cross-compiler and system libraries, use the following command. The `erase=1` will delete temporary files after a successful build.

```
$ make erase=1
```

Host GCC versions 4.7, 4.9, 6.1, and 7.3 are known to work.

After this command completes successfully you have a working cross-compiler for Dreamcast and can compile KOS next.

## Setting up KOS

You should read the documentation in the `kos/doc` directory for details, but here are the basic steps required to set up the KOS environment:

Go into the `kos` directory and copy the template configuration:

```
$ cp /opt/toolchains/dc/kos/doc/envIRON.sh.sample /opt/toolchains/dc/kos/envIRON.sh
```

Now edit `envIRON.sh` to match your installation. If you use the default installation directory you don't need to change anything.

Execute the following command to set the KOS environment variables:

```
$ source /opt/toolchains/dc/kos/envIRON.sh
```

Remember to do this every time you want to use the KOS environment in a newly opened shell. Don't forget to run the above command again when editing `envIRON.sh`.

Now we are finally ready to compile KOS itself. In the `kos` directory, run:

```
$ cd /opt/toolchains/dc/kos  
$ make
```

## KOS-Ports

KOS-Ports is a repository with commonly used libraries for development on the DC, like PNG or MP3 loading.

Clone the repository:

```
$ git clone --recursive git://git.code.sf.net/p/cadcdev/kos-ports /opt/toolchains/dc/kos-ports
```

Compile all KOS-ports using the build-all script

```
$ sh /opt/toolchains/dc/kos-ports/utils/build-all.sh
```

Now you should have a working Dreamcast development environment :-)

Check out the examples in the KallistiOS directory to find out how to use KOS in your own projects!

Retrieved from "[http://dcemulation.org/index.php?title=Compiling\\_KOS\\_on\\_Linux&oldid=3689](http://dcemulation.org/index.php?title=Compiling_KOS_on_Linux&oldid=3689)"

- 
- This page was last edited on 5 March 2018, at 10:15.