Getting Started

To begin building applications on the "HalOS" platform you must first install the Software Development Kit. You can download it from the "HalOS" website www.christopherdeguise.com/halos.

Unpacking the SDK

You should have downloaded a file name HalOS-SDK.tag.gz. You must first unpack the sdk to your home directory with the following commands:

$ gunzip HalOS-SDK.tar.gz
$ tar -xf HalOS-SDK.tar.gz

You should now have a directory named HalOS-SDK.

Setting up the tools:

To build the operating system you need to have GCC and NASM installed. The source code for these tools have been included in the ***tools/*** directory if you do not have them installed, but it will be rare that copies are not already installed on your Linux system.

Building an application

The sample application that is included in the SDK is a simple version of the very popular Tetris, which is used to demonstrate how to build an application. The current method of building your application requires the operating system to be statically linked at build time. The main makefile in the SDK root directory takes care of all the operating specifics, but you will need to add your object dependencies to the list of kernel dependencies as shown below.

Example:

OBJS = kernel/i386strap.o kernel/main.o ….. lib/assert.o *kernel/tetris.o*

Tell "make" what to do to create the object file

Example:

kernel/tetris.o :kernel/tetris.c kernel/tetris.h

These are the only changes that are required in the make file. You will need to alter the entry point of the kernel. This is the file main.c. What main.c does it setup the kernel environment for the application. It then spaws off a process, which is normally your application. You need to add a line like the following, where "tetris" is the function name.

TaskCreate(&Tetris);

Also ensure that you have included the header files for your application as well. Now run make floppy (assuming a floppy disk is inserted). The disk should now boot and run your application.

Make Commands

The main make file contains several directives that allow you to build sections of the operating system.

make clean – Removes all object files and any other temporary files left over from a previous build.

make lib – Builds the C runtime library object files.

make kernel – Build the “HalOS” kernel object file

make boot – Builds the boot loader object files

make floppy – Builds all required object files, creates the disk images and copies it to the floppy disk

make – The default is to build all required object files and create a disk image.

Going Beyond the sample application

All of the source code for the "HalOS" operating system is at your disposal. This means that you can alter it in any way to get it to do what you need. In the sample application you will notice that I have altered the interrupt services routines of COM1 for the app. You can do this too, as that is the power of the system.

The Source

Each source file was named to hopefully mean something useful. The header file and the corresponding .c file have the same name for simplicity. The source code layout is as follows

\HalOS-SDK\ - main directory, includes a Kdevelop project for easy development
- \halos\ - Source code root
  - \boot\ - Boot loader
  - \include\ - Includes files for C runtime and "HalOS"
  - \kernel\ - Kernel source
  - \lib\ - C runtime source
  - \sdk\ - Config files needed to build the SDK using doxygen
  - \testapps\ - Testapplication for the C runtime
- \sdk\ - SDK HTML documentation
- \tools\ - Source code to the tools required to build "HalOS"

If you are attempting to use the halos system, please feel free to email Christopher DeGuise, the author, for assistance at cdeguise@oscillisoft.com.