SoS - Stephen’s Operating System

Stephen Gentle

February 20, 2008

Contents

Chapter 1

Introduction

SoS is a small operating system written in C++ and assembly. SoS supports both the x86 and x86-64 archetectures. This covers the vast majority of desktop and laptop computers available today. Because the GRUB bootloader is used for booting SoS, it can be run off a CD, floppy disk, hard drive or even a USB drive.

In the future, SoS will run a standard C and C++ library, as well as the GNU C compiler and associated tools.

You can download a bootable CD image of SoS, or an archive of the SoS system files for installation onto other mediums like USB drives and hard disks. The GRUB bootloader is recommended for booting SoS, but you can use any multiboot compatible one. There is a floppy disk image of SoS, but we recommend you try one of the other mediums for testing.

Chapter 2

Installing SoS

SoS can be installed on a CD, Floppy disk, USB drive, or hard drive. If GRUB is already installed, then the process is much simpler. You may not have to follow many of the steps below if this is already the case.

SoS will not in any way modify the contents of the media that you install it on, but if you do not install it correctly, you may cause problems with your disks. Because of this, it's best to just burn it to a CD, or use the CD-ROM iso in a virtual machine. Virtualbox and QEMU are the emulators that SoS is mainly tested on, and both are available free of charge on the internet.

2.1 Burning a CD image

SoS is distributed as a CD-ROM iso image. In most operating systems it is very simple to burn an image to a disk, but in some, like Microsoft Windows, you may need third party applications to do this.

If you want to run a modified version of SoS, you can build it from source and have the build system make an image for you. This is not necessary for running the official SoS distribution.

2.2 Installing on a USB drive

Content...

2.3 Installing on a hard-drive

Content...

Chapter 3

Using SoS

If you have SoS installed on a CD or on a USB drive, then to run it you simply have to put it in the drive of your computer, and start it up. When the BIOS starts (the first thing you see when you turn your computer on), it may have something like ’Boot Menu: F12’ displayed either along the bottom of the screen, or in the top corner. Press the key it says, and you should get a list of drives on your computer. If you are booting from CD, select ’CD-ROM Drive’ or ’DVD-ROM drive’ or similar, and press enter. If you are booting from USB, you want ’USB Device’ or the like.

SoS will boot and display the start up log. The terminal is at the bottom of the screen.

3.1 Commands

Blah

Chapter 4

Applications

4.1 Seashell

Seashell will is SoS's default terminal. Right now, the terminal is part of SoS's kernel, but after I have finished porting the C library, and have an elf loader, it will become a seperate application.

Once SoS has a graphical user interface, a graphical version of Seashell will be written.

4.2 GNU Compiler Collection

Once SoS has a C library, I intend to port the whole GNU compiler collection to SoS. This will require a standard C library to be ported first. Then, I will make a cross compiler for my operating system, and compile GCC for it.

This will make it easy to port various UNIX applications (like the nano text editor and Bash) to SoS.

Before this though, I have a lot of work to do - I will have to finish memory management, write a VFS, and write a device driver and filesystem support for CD-ROM drives. I don’t know if I’m going to be able to support USB drives for a long while, unless they act as emulated hard drives.

Chapter 5

Development

SoS is written completly in C++ and assembly. I am trying to keep the code as clean as possible, after previous attempts at writing operating systems that ended up being very dis-organised and messy, and was generally very hard to add new features to and maintain.

5.1 Contributing

Developing an operating system with just one developer is very difficult, and as such, I welcome any contributions. The best way to do this is to download the SVN tree, make your modifications, and then email me the diff.

5.2 Getting the Source

You can download SoS's source code by entering

```
svn co http://s-os.googlecode.com/svn/trunk s-os
```

in your terminal.

5.3 Compiling

To compile SoS, you must first have a GCC toolchain installed. SoS is written and tested with Linux, so that is probably the best way to go, although there's no reason that a cygwin cross compiler wouldn't work. If you're using Ubuntu, you may have to download the build-essentials package using apt-get or Synaptic.

To compile SoS, simply run the command `make` in the SVN trunk directory (the one that contains the makefile and most of the C++ source files). This will build the SoS kernel image and save it in the image folder (in image/system).

To make a CD image for testing either on real hardware, or on an emulator, use the `make install` command.

5.4 Coding Style

Almost everything (except for a few miscelaneous functions like `memcpy()`, and the heap management functions like `kmalloc()`, `kfree()` are organised into classes. There is a different class for each main feature - like the kernel, the GDT and IDT code, keyboard management etc.. The class is defined in a header file with the same name as the class (eg. `gdt.h` contains the GDT class), which is placed in the include directory.

5.4.1 Archetectures

If your code is archetecture specific, remember to enclose it in `#ifdef` and `#endif` directives specifing the platform that the code runs on - for example, `PLATFORM_x86` or `PLATFORM_x86_64`.

5.4.2 Indentation

The style of indentation that I prefer is to put the opening brace on the line below the function / if statement etc. and then start a new tab level. Make sure your text editor adds tabs, not spaces. For example:

```
#include "include/common.h"

void an_example_function(int a)
{
	int c = 7;

	if(a == c)
	{
		// Do something
	}
}
```

Chapter 6

Licencing and Copyright

SoS is Copyright Âľ’ 2003 - 2008, Stephen Gentle.

Portions of SoS are based on code Copyright Âľ’ James Molloy 2007 - 2008.

Portions of SoS are based on code Copyright Âľ’ Brandon F, 2006 - 2008.

SoS is licenced under the BSD licence. For more information, and a copy of the licence, please see the SoS website.