

Getting Started

Kernel development is a lengthy process of writing code, as well as debugging various system components. This may seem to be a rather daunting task at first, however you don't necessarily require a massive toolset to write your own kernel. This kernel development tutorial deals mainly with using the Grand Unified Bootloader (GRUB) to load your kernel into memory. GRUB needs to be directed to a protected mode binary image: this 'image' is our kernel, which we will be building.

For this tutorial, you will need at the very least, a general knowledge of the C programming language. X86 Assembler knowledge is highly recommended and beneficial as it will allow you to manipulate specific registers inside your processor. This being said, your toolset will need at the bare minimum, a C compiler that can generate 32-bit code, a 32-bit Linker, and an Assembler which is able to generate 32-bit x86 output.

For hardware, you must have a computer with a 386 or later processor (this includes 386, 486, 5x86, 6x86, Pentium, Athlon, Celeron, Duron, and such). It is preferable that you have a secondary computer set up to be your testbed, right beside your development machine. If you cannot afford a second computer, or simply do not have the room for a second computer on your desk, you may either use a Virtual Machine suite, or you may also use your development machine as the testbed (although this leads to slower development time). Be prepared for many sudden reboots as you test and debug your kernel on real hardware.

Required Hardware for Testbed

- a 100% IBM Compatible PC with:
 - a 386-based processor or later (486 or later recommended)
 - 4MBytes of RAM
 - a VGA compatible video card with monitor
 - a Keyboard
 - a Floppy Drive
- (Yes, that's right! You don't even NEED a hard disk on the testbed!)

Recommended Hardware for Development

- a 100% IBM Compatible PC with:
 - a Pentium II or K6 300MHz
 - 32MBytes of RAM
 - a VGA compatible videocard with monitor
 - a Keyboard
 - a Floppy drive
 - a Hard disk with enough space for all development tools and space for documents and source code
 - Microsoft Windows, or a flavour of Unix (Linux, FreeBSD)
 - an Internet connection to look up documents
- (A mouse is highly recommended)

Toolset

Compilers

- The Gnu C Compiler (GCC) [Unix]
- DJGPP (GCC for DOS/Windows) [Windows]

Assemblers

- Netwide Assembler (NASM) [Unix/Windows]

Virtual Machines

- VMWare Workstation 4.0.5 [Linux/Windows NT/2000/XP]
- Microsoft VirtualPC [Windows NT/2000/XP]
- Bochs [Unix/Windows]