CHB NOTES AND MANUAL

Version: 0.01 Pre-Alpha

0. Introduction

(C)omputers (H)elp (B)ootloader is a two-stage bootloader
implemented for x86 architecture (x86_64, ix86)

This code implements a bootloader for x86 BIOS systems at the moment
CHB only support loading of binary files (not executables like-elf)

Anyone is FREE to contribute to the CHB project if they are interested. I encourage anyone interested in this project to contribute to it.

1. Building Tutorial

To build CHB bootloader you must have the following tools

1) GCC Cross compiler for x86_64 or ix86 that generates ELF files (I recommend that it be a version greater than 10.1.0)
2) GNU-LD The linker for your compiler, that generates ELF files for x86_64 or ix86 (I recommend that it be a version greater than 2.35)

The makefile doesn't do version checks for the linker and compiler because I don't know what older versions are like, If someone wants to make a port for previous versions of GCC, I encourage them to do so.

Well, if you already have all this you can continue with the tutorial

To build all CHB objects run this in your shell:

```
./compile <your compiler prefix>
```

To see more information about this command try:

```
./compile help
```

When CHB finishes compiling generates the following objects:

1) stage1.IMG: The CHB bootsector, executes the kernel loader
2) loader.IMG: this image contains the kernel loader
3) CHB.IMG: is the union between stage1.IMG and loader.IMG
4) sizes.h: This header file contains the size of the loader so that it can be loaded by stage1, this file should never be edited, it is autogenerated by the size-test.sh script

2. How to Install CHB on raw image file

To install CHB on the image you need to create an empty image with dd (or the command of your preference):

```
dd count=<your number of sectors> if=/dev/zero of=<your image name>
bs=<your sector size>
```

Well, when you have the empty image created copy the CHB.img file to the first sectors of the disk:

```
dd if=CHB.IMG of=<your image name> seek=0 conv=notrunc
```

Well done. You have an image with CHB installed!

Now, you can test the image with QEMU, BOCHS, or another debugger/emulator.

3. Cleaning CHB

For clean CHB output files your try:

```
./clean
```

3. How to Install your kernel on formatted CHB image

CHB by default load the kernel in the fifth sector on the physical address 0x100000, in protected mode, you can change these parameters by editing the CHB code and recompile it.

You need a formatted image of CHB, and copy your kernel in the fifth sector of your image.

```
dd if=<your kernel raw image> of=<your image name> seek=5 conv=notrunc
```

4. CHB NOTES

* Your kernel MUST have compiled for 32-bit platform (iX86 or using -m32 compiler flag) CHB doesn't support long mode!

* NEVER run the makefile directly, because the makefile needs a configuration program beforehand, all of this is provided by the ./configure command. Also note that the ./configure command should never be executed directly either because it will not work to build/clear CHB, this script is only used as a configuration for the makefile.

* Note that the configure script is not the same as the script provided by autoconf command.

5. CHB PLAN (TODO list)

* Add support for filesystems (FAT, ext2)
* Load PE and ELF kernel files in memory
* Add C part in loader
* Fully support for Virtual File System (VFS)
* Load initrd file
* Support for multiboot specification
* Support for UEFI
* Support for non-x86 architectures (mips, arm)
* Support for x86 long mode
* Add command line
* Add support for Hard disk and CDROMS
* Implement full suport for extended disk read mode (INT13,42)