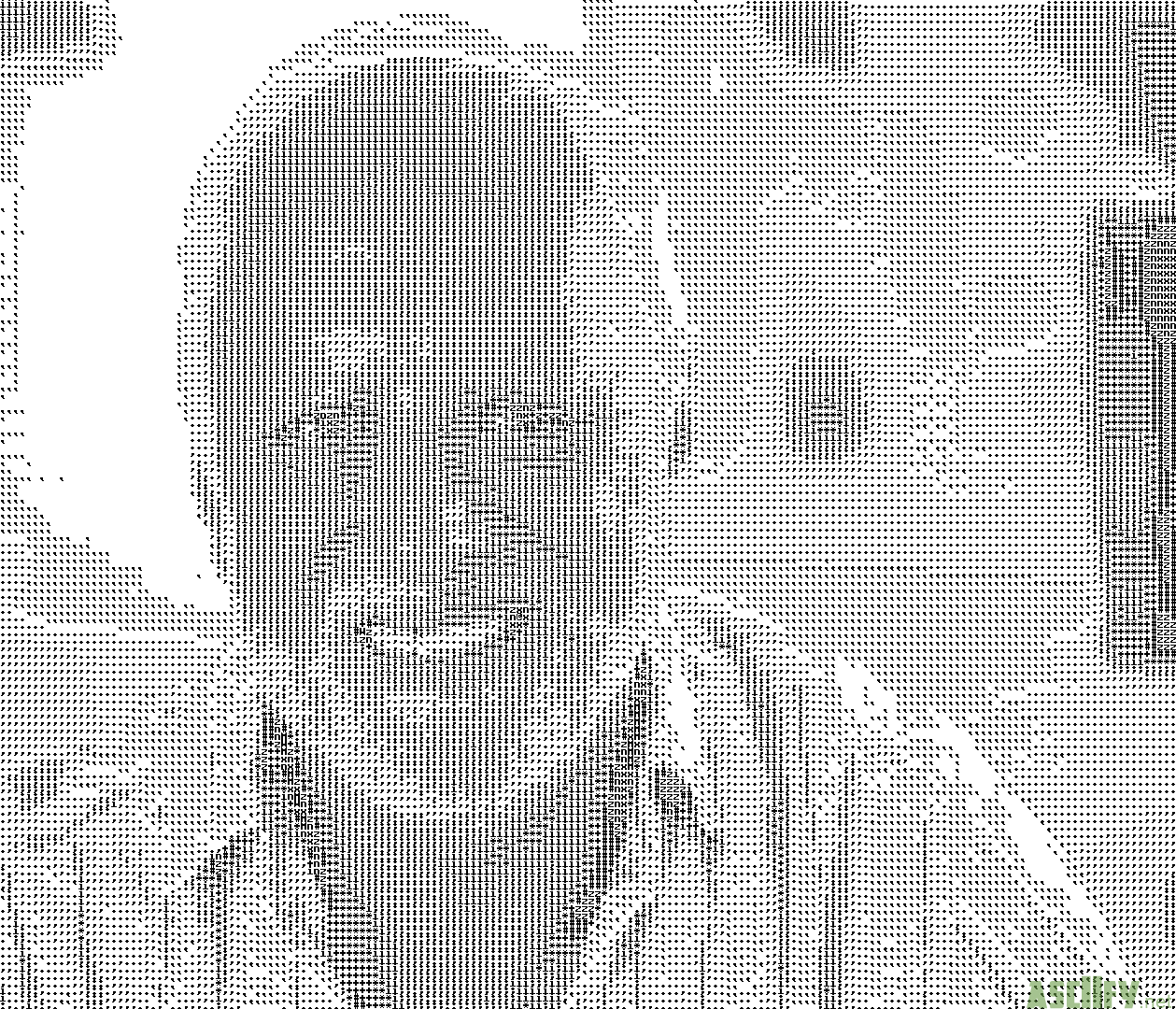
**­­­­**

**H4R0LD OS**

**By Michael Trittin, RJ DeCramer, Jaron Goodman**

**Instructions for Building**

Building is simple. Just run "make all".

However, you don’t need to run this command if you use the provided runOS command, since it will make the project for you and also start bochs.

**Instructions for Usage**

After running our OS using the runOS command, you will be greeted by the patented H4R0LD shell prompt. At this prompt, you can run a variety of commands that showcase the functionality of our operating system. Below we have documented the commands, but you can also access a help menu by using the help command at the prompt.

**Supported Commands**

1. help
2. type [filename]
3. execute [program]
4. copy [in] [out]
5. create [filename]
6. delete [filename]
7. dir
8. execforeground [program name]
9. clear
10. kill [process number]

**Testing the Commands**

help – If it prints the help menu, it’s working

type – Try type messag – if it prints out the message, it’s working.

execute – Try execute phello – if it prints out a bunch of Hello world’s, it’s working.

copy – Try copy messag other, then ‘type other’. If it prints out the message, it’s working.

delete – Try delete messag, then ‘type messag’ and ‘dir’. If the messag doesn’t print out, and ‘dir’ does not display it, then it works.

dir – If it prints out the directory with file sizes, it’s working.

execforeground – Try execforeground phello. Then try typing something (say ‘kill 1[enter]’). You shouldn’t be able to. The input should be buffered and output after the program ends.

clear – If it clears the console, it’s working.

kill – Try execute phello, and then type kill 1[enter] while it’s still running. If the program terminates, then it’s working.

**Bugs or Deficiencies**

1. Currently the behavior of create is to create a new file regardless of pre-existing files with the same name. Thus you can have two files of the same name on the operating system, and can only access the first one created (i.e. the file earlier in the directory). However, you can access the newer file by using the delete command, since it will delete the first file it comes across in the directory.

**Additional Features**

*New in Milestone 5*

11. Added several additional utility commands, such as help and clear.

12. Added an atoi function (only accepts one char)

*Previous Milestones*

1. A cleaner bash script which cleans all irrelevant files from the directory

2. A runOS bash script which generates the necessary kernel/floppy files and runs the OS

3. Organized kernel structure making proper use of header files & splitting up tasks into different files

4. A bash script called linter which runs cpplint.py on each c file in the directory and opens the file in sublime text if there are any errors

5. A bash script called stringMaker which runs the stringMaker program and copies the output to the clipboard for easy access.

6. A few utility functions such as printInt(int i), strcmp(char\* a, char\* b), and strncmp(char\* a, char\* b, int len), zero(char\* buf, int len). These will be very useful for expanding our kernel in the future.

7. All of the compilation/cleaning logic was moved into the Makefile (subsequently removed compileOS and cleaner).

8. Added a bunch of string utility functions (strlen, strpos, etc.)

9. Organized project into directories.

10. Our "dir" command within the shell includes the length (in sectors) of each of the files.