PRJ-1 DUE DATE: Monday Sep 8th, 2008 at 11:59pm
You have 10.5 days to work on project 0
You must work as individuals
PRJ-1 is worth 5% of your overall grade

You will submit your solution in your subversion repository. You should name your project directory `prj-1`*. In addition you will demonstrate your solutions in lab section. I will give you more info on how and what to submit.*

This is a warm-up project to get you familiar with the C Programming Language and some UNIX system calls.

1 hexdump

Write a C/UNIX program called `hexdump` that will dump the contents of a file as a series of lines that show the offset, the byte values in hexadecimal, and the ASCII version of the data. Here is an example.

```
$ cat hello.c
#include<stdio.h>

int main(int argc, char **argv)
{
        printf("Hello World\n");

        return 0;
}
$ ./hexdump hello.c
00000000  23 69 6e 63 6c 75 64 65  3c 73 74 64 69 6f 2e 68  |#include<stdio.h|
00000010  3e 0a 0a 69 6e 74 20 6d  61 69 6e 28 69 6e 74 20  |>..int main(int |
00000020  61 72 67 63 2c 20 63 68  61 72 20 2a 2a 61 72 67  |argc, char **arg|
00000030  76 29 0a 7b 0a 09 70 72  69 6e 74 66 28 22 48 65  |v).{..printf("He|
00000040  6c 6c 6f 20 57 6f 72 6c  64 5c 6e 22 29 3b 0a 0a  |llo World\n");..|
00000050  09 72 65 74 75 72 6e 20  30 3b 0a 7d 0a           |.return 0;.}.|
0000005d
```

Your output should match the format in the example above. For this problem you will need to used UNIX system calls for opening and reading files. You will also need to have a good understanding of the `printf()` C library function. Note that hexdump is a utility that exists in Linux and many other operating systems. While it is easy to find source code on the internet, you should implement your version from scratch.

2 filehist

Write a C/UNIX program called `filehist` that will generate histogram data for the contents of a file. That is, `filehist` should count the number of occurrences of each byte in a file and print the results. Here is an example:

```
$ cat foo
Hello World.
```

```
$ ./filehist foo
0a 1
20 1
2e 1
48 1
57 1
64 1
65 1
6c 3
6f 2
72 1
```

Note that your output should be ordered from lowest existing byte value to highest. You can omit byte values that are not present in the file.

3 minish

Write a C/UNIX program called `minish` that provides minimal shell functionality. Your shell should be able to execute programs and redirect the output of a program to a file. Here is an example:

```
$ ./minish
@ date
Thu Aug 28 13:13:12 PDT 2008
@ cat foo
Hello World.
@ date > mydate
@ cat mydate
Thu Aug 28 13:13:51 PDT 2008
```

In order to implement `minish` you will need to learn the UNIX system calls to start and wait for processes and to fork new processes. You will also have to understand how file descriptors work in UNIX.