COMP 222 Computer Organization
Assignment #2—Cache Memory

Objective:

To simulate reading and writing to a custom-sized direct-mapped cache, involving a custom-sized main memory.

Inputs:

The total size of accessible main memory (in words)
The total size of the cache (in words)
The block size (words/block)
A signal to read (0) or write (1) to the cache
The main memory address to read from/write to
The contents of the address for writing to the cache

Outputs:

The corresponding cache tag, block, and word for a main memory address
The contents of the address resulting from reading/writing to the cache
A message indicating either a hit or a miss to the cache

Specification:

The program simulates reading from and writing to a cache based on choosing from a menu of choices, where each choice calls the appropriate procedure, where the choices are:

1) Enter parameters
2) Access cache for reading/writing and transfer data
3) Quit program

Notes:

- Use a structure (struct) to represent a cache line consisting of a tag (integer) and a block (integer pointer). Define the cache to be a pointer to the struct.
- Upon entering the parameters, the main memory and cache are to be dynamically allocated based on their respective total sizes. Each word *i* of main memory is initialized with the value *M–i*, where *M* is the size of main memory in words.
- Reading/writing from/to a new block in the cache results in dynamically allocating the block based on the block size.

What to turn in:

Softcopy of source code submitted to http://moodle.csun.edu via the submission instructions.

Be sure to name your source code: asmt2_*yourlastname*.c
Any deviation from the format for submission will result in an automatic -10%. You can use any editor and/or compiler, but **make sure** your code compiles and executes under the gcc compiler and the Unix environment—otherwise you will receive 0 points for compilation and execution.

```
% asmt2

Main memory to Cache memory mapping:
--------------------------------------
1) Enter parameters
2) Access cache for reading/writing and transfer data
3) Quit

Enter selection: 1

Enter main memory size (words): 65536
Enter cache size (words): 1024
Enter block size (words/block): 16

Main memory to Cache memory mapping:
--------------------------------------
1) Enter parameters
2) Access cache for reading/writing and transfer data
3) Quit

Enter selection: 2
Select read (0) or write (1): 1

Enter main memory address to write to: 65535
Enter value to write: 14
Write miss!
Word 15 of block 63 with tag 63 contains value 14

Main memory to Cache memory mapping:
--------------------------------------
1) Enter parameters
2) Access cache for reading/writing and transfer data
3) Quit

Enter selection: 2
Select read (0) or write (1): 0

Enter main memory address to read from: 65535
Read hit!
Word 15 of block 63 with tag 63 contains value 14

Main memory to Cache memory mapping:
--------------------------------------
1) Enter parameters
2) Access cache for reading/writing and transfer data
3) Quit

Enter selection: 2
Select read (0) or write (1): 1

Enter main memory address to write to: 65534
Enter value to write: 512
Write hit!
Word 14 of block 63 with tag 63 contains value 512

Main memory to Cache memory mapping:
--------------------------------------
1) Enter parameters
2) Access cache for reading/writing and transfer data
3) Quit

Enter selection: 2
Select read (0) or write (1): 0

Enter main memory address to read from: 1023
Read miss!
Word 15 of block 63 with tag 0 contains value 64513

Main memory to Cache memory mapping:
--------------------------------------
1) Enter parameters
2) Access cache for reading/writing and transfer data
3) Quit

Enter selection: 3
%
```