Project 2 – SHA-256 One Way Hash
EECS 3540: Operating Systems and Systems Programming
Due: Tuesday March 15, 2016

Description:

In this project you are to use the SHA-256 One Way Hash algorithm to compute the hash value for a file. You are to use multiple threads or processes to do this. The input will come from standard input and the hash value should be output (in hexadecimal format) to the standard output.

Details:

The SHA-256 one way hash algorithm computes a 256 bit hash value for a file. The file is broken up into 512 bit blocks and fed into the algorithm. The initial hash value is $H^{(0)}$ and the final hash value is $H^{(t)}$ where t is the number of 512 bit blocks in the file to be hashed.

Step 1: Padding the message. The message will need to be padded out to a multiple of 512 bits. The message length, in bits is, length. Add a "1" bit to the message. Compute the smallest positive k that will be a solution to the equation length + 1 + k = 448 mod 512. This k represents the number of "0" bits that must be added onto the message. Finally add on the 64 bit representation of the message length in bits. This will ensure your message is a multiple of 512 bits.

Computation Steps:

Prepare the message schedule :

for I = 1 to the number of blocks in the message

1. Prepare the message schedule { W_t }:

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{(256)}(W_{t-1}) + W_{t-7} + \sigma_0^{(256)}(W_{t-15}) + W_{t-16} & 16 \le t \le 63 \end{cases}$$

2. Initialize the eight working variables (a, b, c, d, e, f, g, h) with the (i-1)st hash value:

$a = H_0^{(i-1)}$	$b = H_1^{(i-1)}$	$c = H_2^{(i-1)}$	$d = H_3^{(i-1)}$
$e = H_4^{(i-1)}$	$f = H_5^{(i-1)}$	$g = H_6^{(i-1)}$	$h = H_7^{(i-1)}$

3. For t = 0 to 63:

$$T_1 = h + \Sigma_1^{(256)}(e) + Ch(e, f, g) + K_t^{(256)} + W_t$$
$$T_2 = \Sigma_0^{(256)}(a) + Maj(a, b, c)$$
$$h = g$$
$$g = f$$
$$f = e$$
$$e = d + T_1$$
$$d = c$$
$$c = b$$
$$b = a$$
$$a = T_1 + T_2$$

4. Compute the i th intermediate hash value H(i):

$H_0^{(i)} = a + H_0^{(i-1)}$
$H_1^{(i)} = b + H_1^{(i-1)}$
$H_2^{(i)} = c + H_2^{(i-1)}$
$H_3^{(i)} = d + H_3^{(i-1)}$
etc.

Functions:

$$Ch(x, y, z) = (x \,\&\, y)\ xor\ (not\ x\ and\ z)$$
$$Maj(x, y, z) = (x \,\&\, y) xor\ (x \,\&\, z) xor\ (y \,\&\, z)$$
$$\sum_{0}^{(256)}(x) = ROTR^{2}(x) + ROTR^{13}(x) + ROTR^{22}(x)$$
$$\sum_{1}^{(256)}(x) = ROTR^{6}(x) + ROTR^{11}(x) + ROTR^{25}(x)$$

Suggestions:

Use unsigned int's for the message and the numbers in the computation. Do not use signed values.

Work small to start with. I will have a series of sample inputs and hashes for you. There will be a 1 character file with a hash – should be pretty simple. Another one with a few more characters, a full block, and an example with 2 blocks. You may want to print out hash keys between stages to check longer versions.

Write it to work without threads or processes first!

You may use whatever method you wish to communicate between the threads or processes.