Department of Computer Science and Engineering
Chungnam National University

10/2/2010

시스템 프로그래밍 과제 1. 제 2장 김형신

제출일시 : 10월 8일, 금요일 오후 1시 이전까지

과제는 반드시 손으로 작성하여 제출해야 하며(프로그래밍 문제도), 문제풀이 과정이 드러나야 합니다. ( )안의 숫자는 배점을 의미합니다.

1. (2진수의 계산)
다음의 10진수들을 16비트 2진수로 나타내시오.(각 5점)
1) 15 2) 28 3) 69 4) 256

아래의 이진수 덧셈을 수행하시오.

5)
```
  01001011
 +10001001
```

6)
```
  00110011
 +01110011
```

2. is_little_endian 이라는 함수를 작성하시오. 이 함수에서는 little_endian 컴퓨터에서 실행하는 경우 1을 리턴하고, big_endian 컴퓨터에서는 0을 리턴해야 한다. 단, 이 프로그램은 컴퓨터의 워드크기와 무관하게 동작해야 한다. (20점)

3. Practical Problem 2.46 (page 136)

Practice Problem 2.46

The imprecision of floating-point arithmetic can have disastrous effects. On February 25, 1991, during the first Gulf War, an American Patriot Missile battery in Dharan, Saudi Arabia, failed to intercept an incoming Iraqi Scud missile. The Scud struck an American Army barracks and killed 28 soldiers. The U.S. General Accounting Office (GAO) conducted a detailed analysis of the failure [72] and determined that the underlying cause was an imprecision in a numeric calculation. In this exercise, you will reproduce part of the GAO's analysis.

The Patriot system contains an internal clock, implemented as a counter that is incremented every 0.1 seconds. To determine the time in seconds, the program would multiply the value of this counter by a 24-bit quantity that was a fractional binary approximation to $\frac{1}{10}$. In particular, the binary representation of $\frac{1}{10}$ is the nonterminating sequence $0.000110011[0011]\cdots_2$, where the portion in brackets is repeated indefinitely. The program approximated 0.1, as a value x, by considering just the first 23 bits of the sequence to the right of the binary point: $x = 0.00011001100110011001100$. (See Problem 2.51 for a discussion of how they could have approximated 0.1 more precisely.)

A. What is the binary representation of $0.1 - x$?

B. What is the approximate decimal value of $0.1 - x$?

C. The clock starts at 0 when the system is first powered up and keeps counting up from there. In this case, the system had been running for around 100 hours. What was the difference between the actual time and the time computed by the software?

D. The system predicts where an incoming missile will appear based on its velocity and the time of the last radar detection. Given that a Scud travels at around 2000 meters per second, how far off was its prediction?

4. Practice Problem 2.47 (page 141)

Practice Problem 2.47

Consider a 5-bit floating-point representation based on the IEEE floating-point format, with one sign bit, two exponent bits ($k = 2$), and two fraction bits ($n = 2$). The exponent bias is $2^{2-1} - 1 = 1$.

The table that follows enumerates the entire nonnegative range for this 5-bit floating-point representation. Fill in the blank table entries using the following directions:

e: The value represented by considering the exponent field to be an unsigned integer

E: The value of the exponent after biasing

2^E: The numeric weight of the exponent

f: The value of the fraction

M: The value of the significand

$2^E \times M$: The (unreduced) fractional value of the number

V: The reduced fractional value of the number

Decimal: The decimal representation of the number

Express the values of 2^E, f, M, $2^E \times M$, and V either as integers (when possible) or as fractions of the form $\frac{x}{y}$, where y is a power of 2. You need not fill in entries marked "—".

Bits	e	E	2^E	f	M	$2^E \times M$	V	Decimal
0 00 00								
0 00 01								
0 00 10								
0 00 11								
0 01 00								
0 01 01	1	0	1	$\frac{1}{4}$	$\frac{5}{4}$	$\frac{5}{4}$	$\frac{5}{4}$	1.25
0 01 10								
0 01 11								
0 10 00								
0 10 01								
0 10 10								
0 10 11								
0 11 00	—	—	—	—	—	—		—
0 11 01	—	—	—	—	—	—		—
0 11 10	—	—	—	—	—	—		—
0 11 11	—	—	—	—	—	—		—

5. Problem 2.85

Intel-compatible processors also support an "extended precision" floating-point format with an 80-bit word divided into a sign bit, $k = 15$ exponent bits, a single *integer* bit, and $n = 63$ fraction bits. The integer bit is an explicit copy of the implied bit in the IEEE floating-point representation. That is, it equals 1 for normalized values and 0 for denormalized values. Fill in the following table giving the approximate values of some "interesting" numbers in this format:

	Extended precision	
Description	Value	Decimal
Smallest positive denormalized		
Smallest positive normalized		
Largest normalized		