CPEG-323: Introduction to Computer Systems Engineering
Handout: Homework assignment 4
Issued: Wednesday, Oct. 15, 2008
Due Date: Monday, Oct. 27, 2008

Instructions

Please begin your answer to every problems on a new sheet of paper. Be as concise and clear as you can. Make an effort to be legible. To avoid misplacement of the various components of you assignment, make sure that all the sheets are stapled together. You may discuss problems with your classmates, but all solutions must be written up independently.

This homework will help you to advance your ability to apply knowledge in computer engineering learned in the course and Knowledge of related topics in computer science discipline.

Problem 1 (25 points)
Please give the definitions for the following items.

(a) performance

(b) latency

(c) "wall clock" time

(d) Weighted CPI

(e) System time

Problem 2 (20 points)
Please briefly answer the following questions.

(a) In order to get accurate performance result, people should measure CPI at run time (dynamically) instead of at compile time(statically). Why?

(b) Please explain why hardware/software designers sometimes still need to measure CPI statically?

Problem 3 (20 points)

Do the Problem 4.10 in Patterson and Hennessy's textbook(see page273).

Problem 4 (35 points)
Note: This question requires a working SimpleScalar installation.

Given the following program:

```
#include <stdio.h>

int f = 11;
int bar(int a, float b, char c){
   int d;
   d = a + (int)((b) / (int)(c));
    return d;
}
int main(int argc, char **argv){
   static int e;
   e = bar(10, 153.3, 'a');
   printf("%d - %d\n", e, f);
   return 0;
}
```

Given all variables used in this program (a, b, c, d, e, f, the return value of bar and its arguments, printf's arguments, and main's return value and its arguments), point out where they are allocated in the ELF image. This means if they are allocated in the bss, data, rodata, stack, registers, etc. Your answer should be in the form of: (variable name, section name or stack offset or register name, size) For variables located in the stack and the registers, you should read the assembly of the file and for the rest; you need to use the nm or objdump utilities.

Finally, please provide a print out of the program's result.

Note: For the string in the printf function, use its assembly name.

Hint: Compile the program with no optimizations

Hint: For complete instructions for compiling and generating the assembly, please refer to the handout provided in Lab1.