

Homework Set 2

CMCS 22620/32620, Spring 2004

Assigned: April 14, 2004

Due: April 21, 2004

1. Instruction selection

- (a) In your own words (in one paragraph), explain the difference between *optimal* and *optimum* tiling of trees. (Hint: Think “local” vs. “global.”)
- (b) Make up an example where an optimal tiling is not optimum.
- (c) Is it possible (in principle) that an optimum tiling is not optimal?
- (d) Argue convincingly (in one paragraph or so) that a correctly implemented **Maximal Munch** will yield an optimal tiling. (Hint: This is best done using an indirect proof. First say what “correctly implemented” means for Maximal Munch, then show that not getting an optimal tiling implies that the Munch was not implemented correctly...)

2. Stack frames

- (a) The *stack pointer* always points to the current stack frame. Certain calling conventions also use a *frame pointer* which points to the caller’s stack frame. If we had a frame pointer on the PowerPC, access to what kind of data would be more convenient to implement? (Hint: Which data “belongs” to the callee despite being allocated in the caller’s frame?)
- (b) Calling conventions for some programming languages use a so-called *static link*. What is the difference between the static link and the dynamic link (or, for that matter, the frame pointer)? Where is the static link pointing?
- (c) Briefly explain the language feature (not present in MiniJava, but present in, e.g., Pascal) whose implementation often uses static links? Argue why dynamic links (i.e., pointer’s to the caller’s stack frame) are not sufficient for implementing the same feature. (Hint: Sketch out a simple example.)