

CS 577 Homework 6 – Integrating Interpretation and Compilation – due 10am, Thursday, May 22, 2008

The goal of this assignment is to integrate an interpreter and a true JIT compiler into a single executable that initially interprets each method, but JIT-compiler it after it has been executed a specified number of times, and thereafter runs the compiled code. You are provided with a working interpreter (a slightly modified variation of the solution to homework 1) and a true JIT compiler (a modified version of the solution to homework 3), which delays compilation until just before a method is first executed. The two mechanisms share a single `main` program; the desired mode of execution (interpret or compile) is specified as an initial command line argument.

Your combined system should behave as follows:

- Instead of a simple execution mode flag, the command line specifies the number n of interpretive calls permitted to any one method.
- When interpreting a call:
 - If the target method has already been compiled, call the compiled code.
 - If not, but the method has already been called n times, compile it now, and then call the compiled code.
 - Otherwise, call the interpreter recursively to execute the method in the existing way.
- When compiling a call:
 - If the target method has already been compiled to native code or has a trampoline, generate a call to that code.
 - Otherwise, compile a trampoline for the target method, and generate a call to that trampoline.
- Trampolines should check if the target method has already been called n times.
 - If so, they should compile it now, rewrite themselves (and the caller) and call the compiled code in the existing way.
 - If not, they should interpret the target method.

Note that the `main` method should be compiled if $n = 0$ and interpreted otherwise.

As in the existing JIT, you should ensure that once a method has been compiled to native code, all subsequent calls to that method invoke the native code as expeditiously as possible. Remember to consider, and generate java test cases for, all possible combinations of caller/callee state (interpreted or compiled).

The trickiest part of the existing JIT code is that it when the trampoline compiles a method it updates its own *caller* to call the resulting native code directly next time it is executed. Note that this is the *wrong* thing to do when calling from the interpreter! The easiest way to avoid problems is to maintain an invariant that the interpreter never calls into a trampoline—only into the body of a compiled method. That way, you don't need to mess with the tricky trampoline code at all. But be sure you understand the consequences of maintaining this invariant!

How to submit your homework

Submit the homework *by email* to `apt@cs.pdx.edu` with the subject line “CS577 HW6” prior to the beginning of class on the due date. You should submit a revised version of files `jit2.c`, `interp2.c`, and `main2.c`, together with any other new or changed files, as *attachments* to your mail.