p3q1
AM

Compiler Construction 2002

**Compiling Techniques y2002p3.tex**

Give a diagram showing the phases of a typical compilation system for a language like C which produces a directly executable fully-linked binary file as output. For each phase, describe in a paragraph what it does (mentioning possible implementation techniques) and give a brief overview of the data-structures used for its input and output indicating whether they would normally reside in a file or in memory. (Do not specify details of any files used to automate the writing of any of the above phases.) [16 marks]

Indicate how a typical Java implementation might differ and explain what is meant by *just-in-time compilation.* [4 marks]

**Solution Notes y2002p3.tex**

| Phase: | input | output |
|---|---|---|
| lex | stream of chars | stream of tokens (in memory) |
| syn | stream of tokens | parse tree (in memory) |
| optional(typechk) | parse tree | annotated parse tree |
| trn | parse tree | stack-machine code (in memory) |
| cg | stack-machine code | target-machine code (as ELF file) (or .asm file) |
| optional(assemble) | target-code as .asm | target-machine code (as ELF file) |
| link | ELF (aux ELF library input) | ELF (fully resolved) |

Java (at least Sun's Java) would differ in that the output of trn would be JVM code which is written to a .class file. Linking a .class file into a program which uses it is done at run-time. The JVM may typically be interpreted, but an option is JIT compilation in which the JVM is translated into target machine code immediately following loading the .class file.