

CSE 401
Intro Compilers

Final Review

(mostly post-Midterm)

Larry Rizzo
Spring 2001

© 1998 UW CSE

1

Compile- vs Run-Time

- procedures vs activation record/stack frame
- scope/symbol table vs environment/stack frame
- variable vs memory/stack/register location
- lexically enclosing scope vs static link
- caller vs dynamic link

2

Run Time Storage

- layout of data structures
- memory areas: static, stack (lifo), heap
- layout of stack frame: formals, locals, links, etc.
- calling conventions
- parameter passing modes: call-by-value vs call-by-reference vs ...

3

Intermediate Code Gen

- Why? How different from target? (temps, machine (in)dependence, ...)
- 3-address code
- gen IR from AST: l- vs r-value, exprs, assign, arrays, ...
- Short circuit code

4


Target Code Gen

- Instruction selection (RISC/CISC)
- Register allocation
- Code scheduling

5

Optimization

- Deduce as much as possible at compile time about run time bindings, values, control flow, ...
- Use it to:
 - Simplify/specialize unnecessarily general code
 - Reorder code
 - Exploit target machine
- Scope:
 - Peephole
 - Local
 - Global (intra-procedural)
 - Inter-procedural



6