Compilers: History and Context

COMP 3002

Carleton
UNIVERSITY
Canada's Capital University

Outline

- Compilers and languages
- Compilers and architectures
  - parallelism
  - memory hierarchies
- Other uses of compilers
  - Program translations
  - Software productivity tools

Programming Language Evolution

- New generations of programming languages have introduced new challenges
  - C, Fortran
    - Register allocation, aggregate types
  - Simula, C++
    - Virtual method dispatch
  - Java and the JVM
    - Bounds checking, type-safety
    - Garbage collection
    - Just-in-time compilation

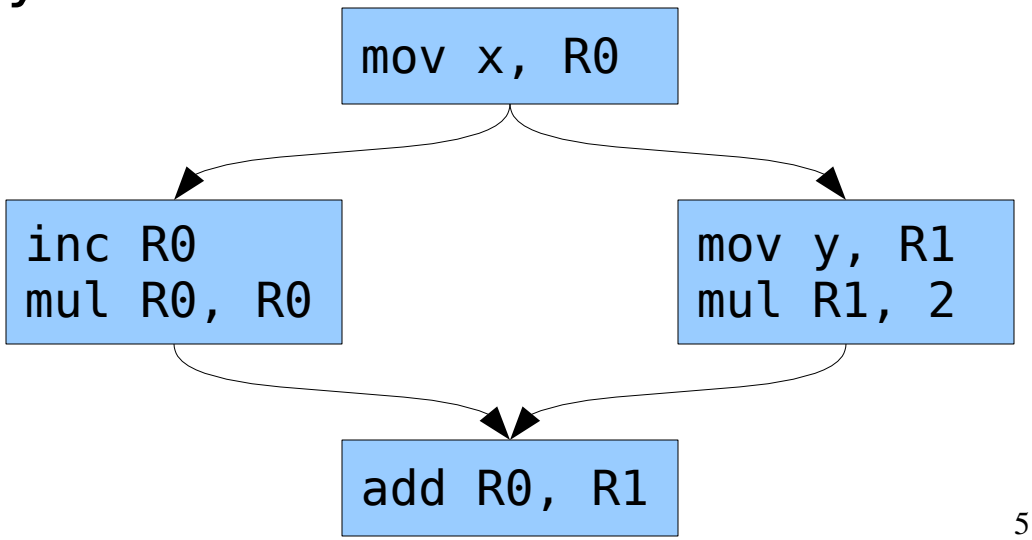
Hardware Evolution

- New types of hardware have introduced new challenges
  - Parallelism
    - Instruction level
    - Multiprocessor
  - Memory hierarchies

Instruction-Level Parallelism

- Modern machines don't execute code sequentially
  - Processor analyzes code to look for instruction dependencies
  - Independent instructions are executed in parallel
- Compilers try to maximize the available parallelism

```
mov x, R0
mov y, R1
inc R0
mul R1, 2
mul R0, R0
add R0, R1
```



Carleton UNIVERSITY
Canada's Capital University

Vector Parallelism

- Many machines now have vector instructions
  - Can load and operate on an entire vector

```
for (i = 0; i < n; i++) {
  a[i] = b[i] + c[i];
}
```

```
for (i = 0; i < n; i += 8) {
  a[i,..,i+7] = b[i,..,i+7] + c[i,..,i+7];
}
for (i -= 8; i < n; i++) {
  a[i] = b[i] + c[i];
}
```

Multiprocessor

- Currently, many machines have multiple processors or cores
  - Compilers can try to to automatically parllelize code
  - We're still not quite there yet

```
do in parallel {
  sum(a, b, c, n/2);
  sum(a+n/2, b+n/2, c+n/2, n-n/2);
}

sum(a, b, c, n) {
  for (i = 0; i < n; i++) {
    a[i] = b[i] + c[i];
  }
}
```

Memory Hierarchies

- Current machines have a multi-level memory hierarchy
  - CPU cache
  - L1 cache
  - L2 cache
  - RAM
  - Swap space
- Managing this hierarchy is like register allocation

RISC

- Reduced instruction set computing
- Success of compilers led to processors having fewer instructions
  - that execute faster
- Complex instructions were to make assembly programming easier

Program Translations

- Compilers
  - Input language A -> Output language B
- Includes
  - binary translation
    - 68040 -> PowerPC
  - hardware synthesis
    - RTL -> hardware design
  - database query interpreters
    - SQL query -> query program
  - compiled simulation
    - simulation spec. -> simulation program

Software Productivity Tools

- Syntax highlight
- Code (re)factoring
- Type checking
- Bounds checking
- Memory-management tools

Carleton
UNIVERSITY
Canada's Capital University