CS480
Translators

Introduction to Compilers
Milestone I - gforth
Your First Milestone

• Learn a new language, *gforth*
• Get a Makefile working
• Write a Milestone report
Why learn a new language?

• Different language paradigms
• This is our target code
What is Forth?

• Imperative
• Stack-based
• Postfix (Reverse Polish Notation)
  1 + 2 * 3
  1 2 3 * +

• How about 3 / 1 + 4 * 2?
Gforth

- How does it work?
  - Command Line
  - Files
Old Desk Calculators

```
flip3 ~/cs480_files 68% dc
1
2
+
p
3
1 2 +
p
3
p
p
p
p
p
dc: stack empty
quit
flip3 ~/cs480_files 69%
```
Starting/Using Gforth

```plaintext
flip3 ~/cs480_files 58% gforth
Gforth 0.7.0, Copyright (C) 1995-2008 Free Software Foundation, Inc.
Gforth comes with ABSOLUTELY NO WARRANTY; for details type `license'
Type `bye' to exit
3 1 / 4 2 * + ok
.11 ok
1 ok
2 ok
3 ok
* ok
+ ok
.s <1> 7 ok
s" Hello everyone" ok
.s <3> 7 32538112 14 ok
bye
```

OSU Oregon State University
Makefile Example

• Four Targets
  – compile
  – clean
  – stutest.out
  – proftest.out
Example stutest.out

Exercise 1:
printf('Hello World\n');
Hello World

Exercise 2:
15 * 3 + 4 - 10 / 2 - 7
37

Exercise 3:
15.0 * 3.0 + 4.0 - 10.0 / 2.0 - 7.0
37.

Exercise 4:
15.0e0 * 3.0e0 + 4.0e0 - 10.0e0 / 2.0e0 - 7.0e0
37.

Exercise 5:
15 * 3.0e0 + 4.0e0 - 10 / 2.0e0 - 7
37.

Exercise 6:
y = 15;
x = 3.0e0;
y * x + 4.0e0 - 10 / 2 - 7
37.
Milestone Report

Handwritten Answers to Milestone Questions:

Specification (what do you think the purpose of this milestone is)

Processing (how did you and/or your team go about solving the problem)

Testing Requirement (how did you and/or your team test for correctness)

Retrospective (what did you learn in this milestone)

Team Evaluation (what is the percentage of time contributed by each team member)
Milestone Grading

• Compiles, cleans, and produces stutest.out
  – 50%

• Testing: completeness of coverage
  – 25%

• Milestone report (done individually)
  – 25%

• Remember to demo within 7 days of the due date!!!