The Problem

Definition

Every engineering problem begins with a need that winds up driving the solution. Engineering is the art of finding a feasible solution to the problem that meets all of the constraints (be they cost, resources, durability, etc.).

- Engineering is the study of trade-offs
- Techniques for managing the complexity apply across domains
- The right solution can be elegant, but is often only visible only in hindsight
The Cat
The Cat
The Door
The Problem

• The cat needs to be indoors by nightfall
• Getting him in during the evening can be difficult
  – Chasing the cat
  – Bribing him with food
  – Not always successful
• Cat comes in approximately every two hours during the day
The Solution

- Need to make the cat door one-way, so the cat can reenter the house, but not leave again
- Set it to one way at 5PM, cat will be in by 7PM (before darkness)

[ ] Don’t have to be home to get the cat in ]
Commercial Solution

• Try to go for a commercial solution
• Buy rather than build
• Not exactly what you need, but can perhaps be made to work.

• “Hav-a-Hart” live animal trap.
Commercial Solution
Commercial Solution
Commercial Solution
Need a new approach

FAILURE!
New approach: Inside Door

- Add a second door inside the first on the wall
- Hinged up for normal operation
- Down to form a “double seal” or one way valve for the cat door
- Larger than opening, cannot be pushed through.
Inside Door
How would I make a product?

Simple two “paw” button action to select modes
Cat and moon shaped lights to indicate various modes
Porch sensor reads only your cat’s microchip number
Magnetic latch to stop the flap opening in the wind

$350
Robust In-Only Option

- Use the existing guillotine door rails
- Allow the cat entry, but block the door from opening to the outside
- Needs to be robust to clever cat
- Needs to be accepted by cat
  - Doesn’t work if cat won’t use it
- Will need some refinement to get right
The Prototype

[Images of a cardboard prototype with tape and a note saying "Bola's one way door"]

[Another image of the prototype installed in a wall]
The Prototype in Action
One-way Action Confirmed

- Cat can get in
- Cat cannot get out

- Prototype successful, move on to better implementation
Prototype version 2
Need to refine prototype

FAILURE!
Prototype version 3
Happy Cat!
Inspiration and Iteration are two very necessary parts of finding your way to a solution that works well. Failure is the genesis of further experimentation, which leads to better design.

- Fail early and often
  - Early reduction of less promising ideas
- Be flexible
  - Don’t get married to the first solution you try
  - Be ready to jettison something that isn’t working

Experimentation leads to more understanding
Questions?
Extra Credit Lab: Due Wednesday the 31st

CMPE-013/L

Introduction to “C” Programming

Maxwell James Dunne
gcc
gcc (GNU C and C++ compiler)

- First compiler for GNU and adapted by many operating systems
- MPLABX is calling a customized variant of GCC to generate the hex file
- Supports both C and C++
gcc
Basic Usage

Syntax

```
gcc -o outfile source files
```

- `-o` Sets the executable output name
  - Without argument defaults to a (a.exe on cygwin)
- **Source files** the set of source files to compile.
  - Example
    - `gcc SimpleMain.c`
    - `gcc -o mml mml_tester.c MatrixMath.c`
gcc
Object Files

• Object files allow individual compilation of source files (skips the link step)
  – Generally not runnable but have machine code for the source file
• In large projects this is essential as full compiles can take hours to complete
• .o files can then be compiled together without the flag as normal.
HTTP://XKCD.COM/303/

THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
“MY CODE’S COMPILING.”

HEY! GET BACK TO WORK!

COMPILING!

OH. CARRY ON.
gcc
Object File Creation

Syntax

```
gcc -c source_files
```

- `-c` Tells the compiler to skip linking
- *Source files* the set of source files to compile into an object
  - Each file will generate a different `.o` file
  - Example
    - `gcc -c mml_tester.c`
make
make

- Command line tool designed to make the process of compiling code easier
- Parses a makefile to determine which actions to take
- MPLABX generates a makefile with the project and that is called when the hammer button is clicked
- Incredibly powerful tool as complex as C itself, will only cover the very basic commands
Invoking make

- Simply type “make” on the command line
- Make will attempt use a file called makefile and process targets from it
- If called without arguments it attempts to run the target all
- With arguments it attempts to create that specific target
- Smart enough to only execute targets that need updating
make
makefile contents

target: dependencies
actions to make target

newtarget: dependency1 dependency2
more actions

25 minutes per compile
Clean build 30 seconds
Selective Compilation with make