Discussion 2

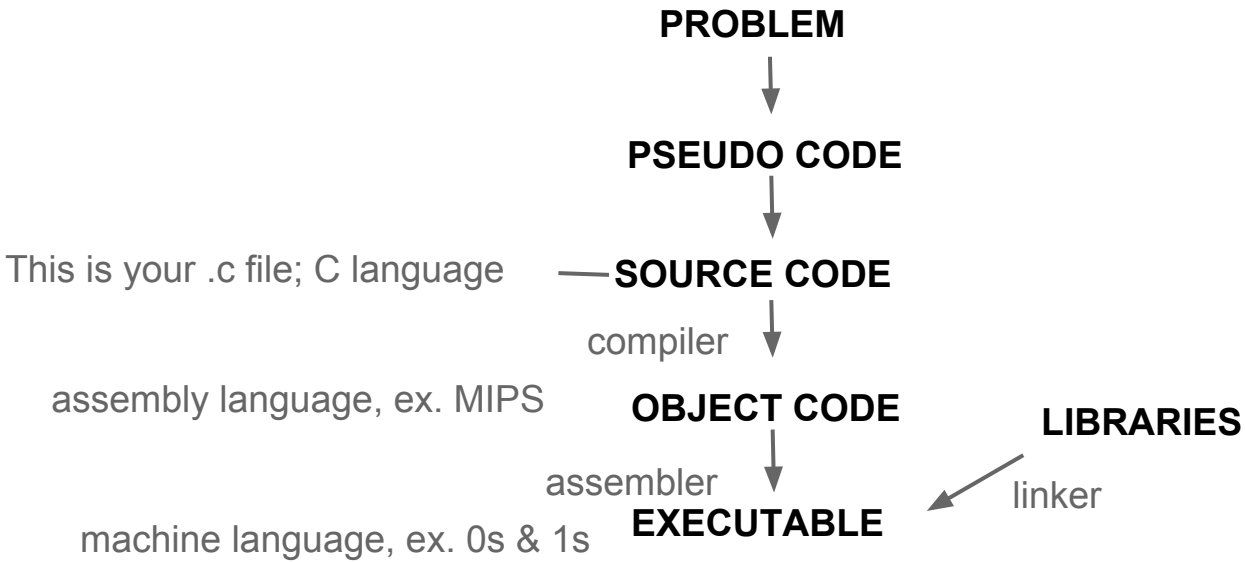
General Annoucements

- Written assignment guides have been posted to Smartsite. Please use to complete and submit your assignments.
- Guides will also be provided for the projects
  - Project 1 to be released today
- If you are working from home (NOT ssh using MobaXTerm or Terminal)
  - https://www.youtube.com/watch?v=HZknU42e8qw

What to do if things don’t work

- Don’t panic (ie. start on your work early)
- Google for the error or problem
- Email the TA’s
  - We will not look through pages of code over email. Send a short snipet and copy of the error message.
- Go to CS Club tutors or TA office hours

Programming Flow Chart



Linking Libraries

- For this class:
  - gccx myfile.c -o myfile
  - (same as) gccx -o myfile myfile.c
- In general:
  - gcc -I /usr/local/lib64/ERoberts <path to library file> -o myfile myfile.c
- we have installed a script for this class to do the linking for you if you use gccx

Types

- int, float, double
  - signed vs unsigned int
  - if int is 16 bits, can represent these numbers:
    - signed -> -2^15 to (2^15) - 1
    - unsigned -> 0 to (2^16) - 1
- char, string
- bool
- ….

Numbers

- Variables:
  - int x = 2;
    - ‘int’ is the type
    - ‘x’ is the name
    - ‘2’ is the value
  - Declare vs initialize/define
    - int x; → declare variable
    - x = 2; → define/initialize variable
  - All ints are truncated (5/3 = 1) (soln: type casting)

Modulo

- The ‘mod’,‘%’ operation is the remainder of x mod y when x is integer divided by y (x/y)
- Ex:
  - 0 mod 2 = 0
  - 1 mod 2 = 1
  - 2 mod 2 = 0
  - 3 mod 2 = 1
  - 4 mod 2 = 0
  - …..
- 0 mod 3 = 0
- 1 mod 3 = 1
- 2 mod 3 = 2
- 3 mod 3 = 0
- 4 mod 3 = 1
- 10 mod 3 = 1
- 200 mod 50 = 0
- x mod 1 = 0
  - anything mod 1 is 0

Loops and Conditional Statements

- for loop:
  - syntax:
    - for(initialize; condition; increment/decrement)

```
{
  expression;
  expression;
 }
```

Loops and Conditional Statements

- for loop:
  - example:

```
int i;
for (i = 0; i < 10; i = i + 1)
 {
    printf(“%g ”, i);
 }
```

  - result will be: 0 1 2 3 4 5 6 7 8 9

Loops and Conditional Statements

- while loop:
  - syntax:
    - while(condition)

```
{
  expression;
  expression;
 }
```

Loops and Conditional Statements

- while loop:
  - example:

```
int i = 0;
while (i < 10)
{
   printf(“%g ”, i);
        i = i + 1;
}
```

  - result will be: 0 1 2 3 4 5 6 7 8 9 (same as for loop)

Loops and Conditional Statements

- if/else conditional statement:
  - syntax: (if, if/else, if/else if/else)
    - if(condition)

```
{
  expression;
}
else
{
  expression;
}
```

Loops and Conditional Statements

- while loop with if/else:
  - example:

```
int i = 0;
while (TRUE)
 {
  if (x == 10)
        break;
  else
        {
         printf(“%g ”, i);
           i = i + 1;
        }
  }
```

- result will be: 0 1 2 3 4 5 6 7 8 9 (same as for loop and previous while loop)