Programs that reference their own descriptions

Print out two copies of the following, the second one in quotes: “Print out two copies of the following, the second one in quotes:”

Print out this sentence.

ECS 120 – 5/24/2013

With thanks to slides of John Black (U of Colorado, Boulder) and (Stanford)
Program that prints $<B>$

Program that prints the description of the program that prints $x$

$<A>$ On input $x$:

\[ \text{PRINT} <B> \]

$<B>$ On input $x$:

print \( \text{PRINTPROG}(x) \parallel <x> \)

$<A>$ $<B>$
char*f="char*f=%c%s%c;main(){printf(f,34,f,34,10);}%c";main(){printf(f,34,f,34,10);}

• A self-replicating program (a “quine” (pronounced “kwine”))

• Using format f in a printf statement, it prints f
  – Yow!

• **Lesson**: you can write (weird) programs that
  – Know their own description, and
  – Use that description in any arbitrary way. In effect,
  – Can say: “obtain your own description” in pseudocode

**login.c** – Trojan login
Like real `login.c` but let’s me on to any account if I give password *LetMeIn! 6!*

**cc.c** – Trojan C compiler
Like the real C compiler `cc.c` except:

- Compile `login.c` to `login = cc(login.c)`
- Compile `cc.c` to `cc = cc(cc.c)`

How does `cc.c` “know” `cc.c`?
See “Lesson”!

1. Write `login.c` and `cc.c`
2. Compile with `cc` to `login` and `cc`
3. Throw away `login.c` and `cc.c`
4. However often you recompile `login.c` and `cc.c`, you get `login` and `cc`