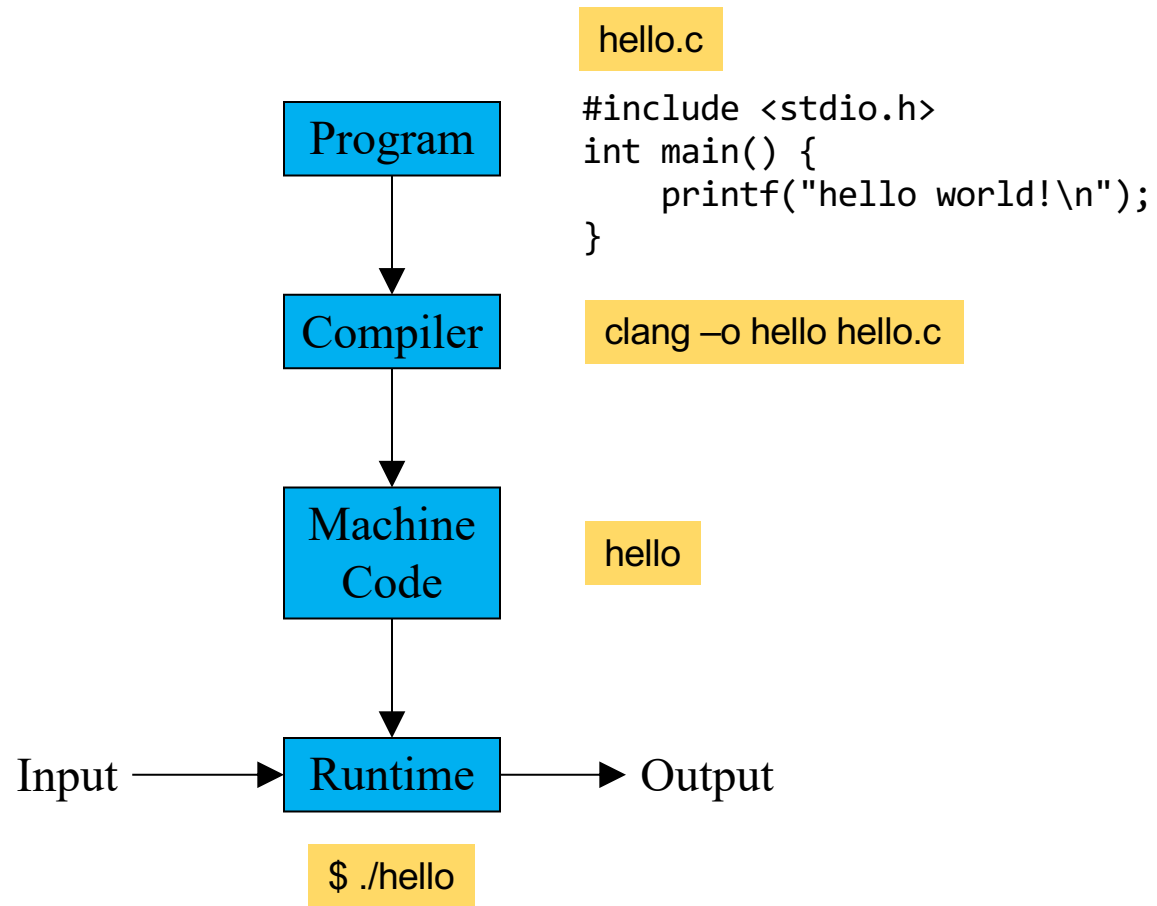


Introduction to Compilers

CMPT 379: Compilers

Instructor: Anoop Sarkar

anoopsarkar.github.io/compilers-class



Program

What is a program?

hello.c

```
#include <stdio.h>
int main() {
    printf("hello world!\n");
}
```

```
$ file hello.c
hello.c: c program text, ASCII text
```

ASCII character set

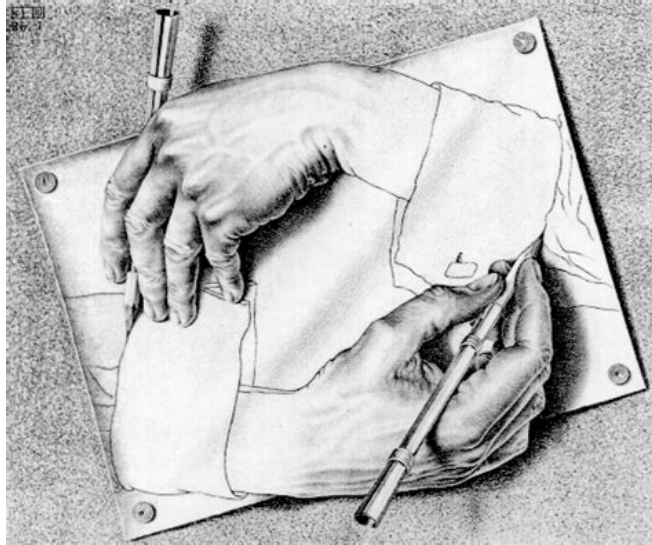
0 nul	1 soh	2 stx	3 etx	4 eot	5 enq	6 ack	7 bel
8 bs	9 ht	10 nl	11 vt	12 np	13 cr	14 so	15 si
16 dle	17 dc1	18 dc2	19 dc3	20 dc4	21 nak	22 syn	23 etb
24 can	25 em	26 sub	27 esc	28 fs	29 gs	30 rs	31 us
32 sp	33 !	34 "	35 #	36 \$	37 %	38 &	39 '
40 (41)	42 *	43 +	44 ,	45 -	46 .	47 /
48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7
56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G
72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [92 \	93]	94 ^	95 _
96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g
104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {	124	125 }	126 ~	127 del

Q: Why 128?

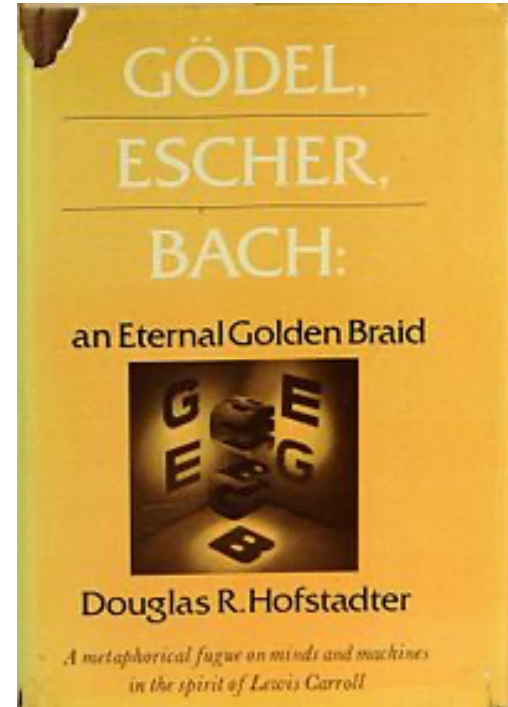
A Quine is a program that generates its own code

- A program is just a text ASCII file
- `printf` prints out ASCII text
- There must be a program that can print out ASCII text that is itself source code for a program
- This would be a program that is a program generator
- A program generator that generates itself is called a Quine

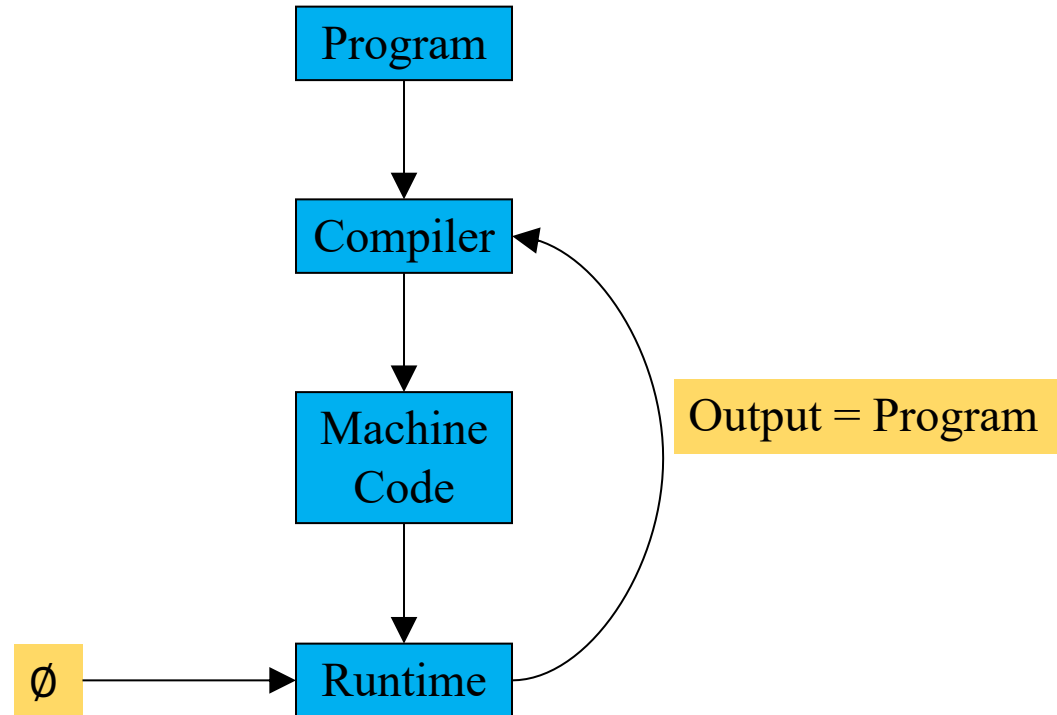
A Quine is a program that generates its own code



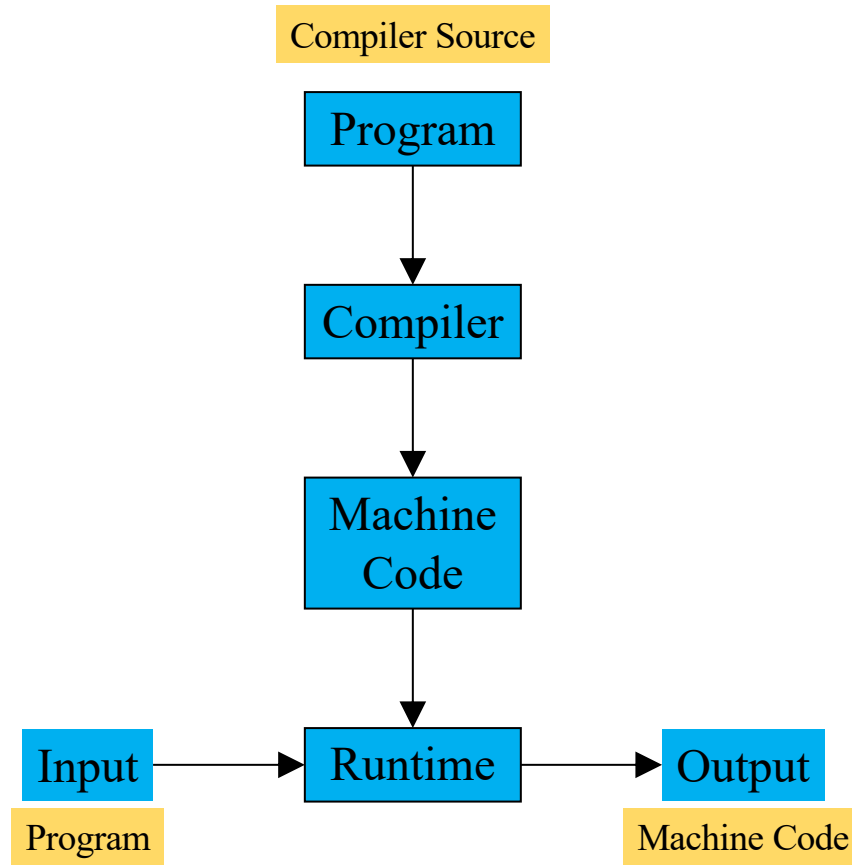
M.C. Escher. "Drawing Hands"



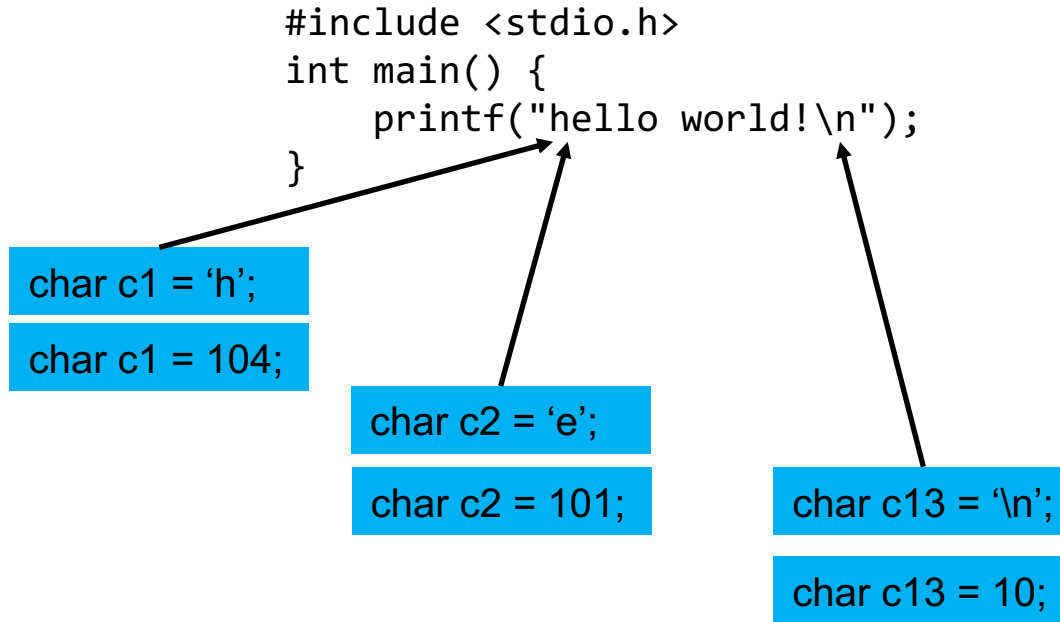
```
#include <stdio.h>
int main(){char *c="#include <stdio.h>%cint main(){char
*c=%c%s%c;printf(c,10,34,c,34,10);}%c";printf(c,10,34,c,34,10);}
```



The compiler has source code – must be compiled



Character constants in programming languages



```
c = next();  
if (c == '\\') {  
    c = next();  
    if (c == 'n')  
        return('\\n');  
}
```

Compiler Source

Program

Compiler

ERROR: '\\n' not a valid character

Machine
Code

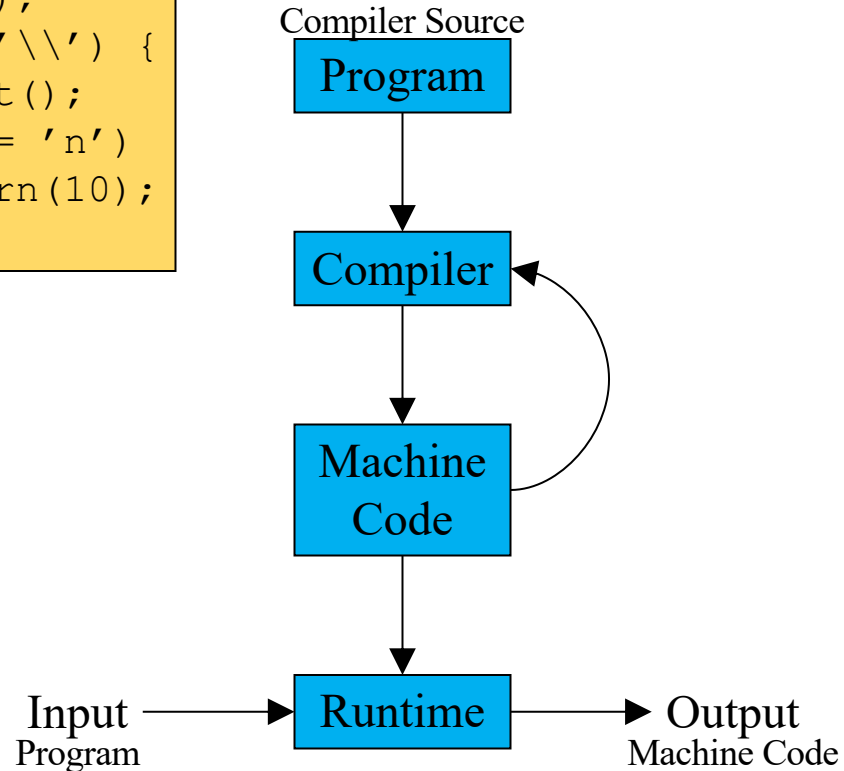
Input
Program

Runtime

Output
Machine Code

```
printf("hello world\\n")
```

```
c = next();  
if (c == '\\') {  
    c = next();  
    if (c == 'n')  
        return(10);  
}
```



```
printf("hello world\n")
```

```
c = next();  
if (c == '\\') {  
    c = next();  
    if (c == 'n')  
        return('\\n');  
}
```

Compiler Source

Program



New
Compiler



Machine
Code



Input
Program

Runtime

Output
Machine Code

```
printf("hello world\n")
```

login is just another program

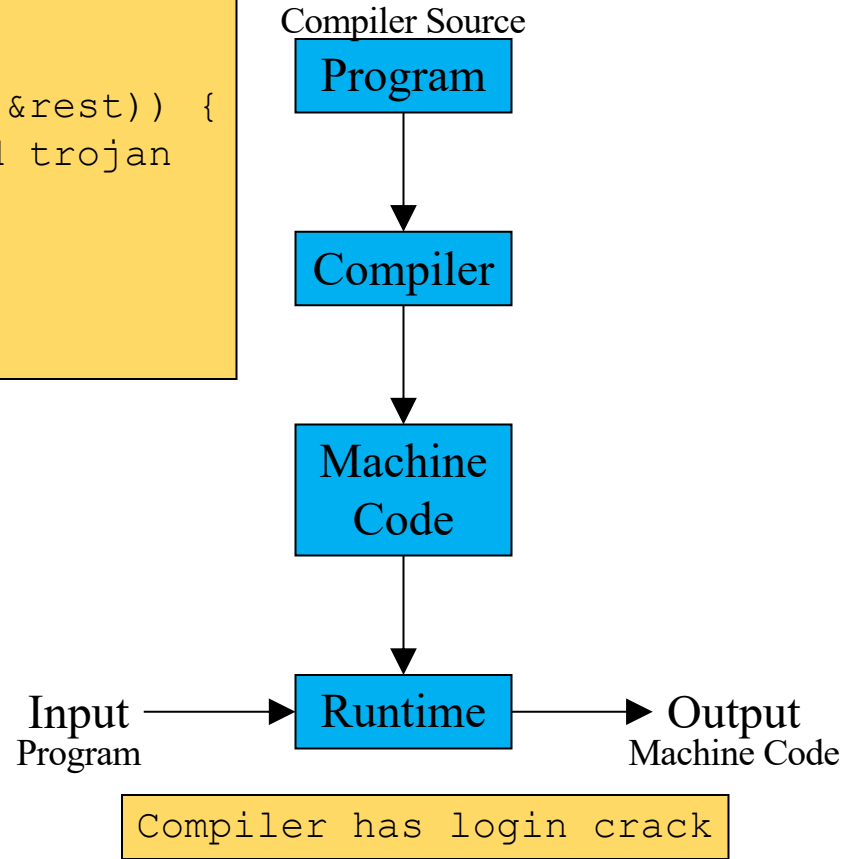
login code from the freebsd GitHub repository:
<https://github.com/freebsd/freebsd>

```
static void
do_login(const struct passwd *pwd, char *tty, char *ttyn)
{
    ...
    struct spwd *sp = getsppnam(pwd->pw_name);
    check_shadow(pwd, sp);
    ...
}
```

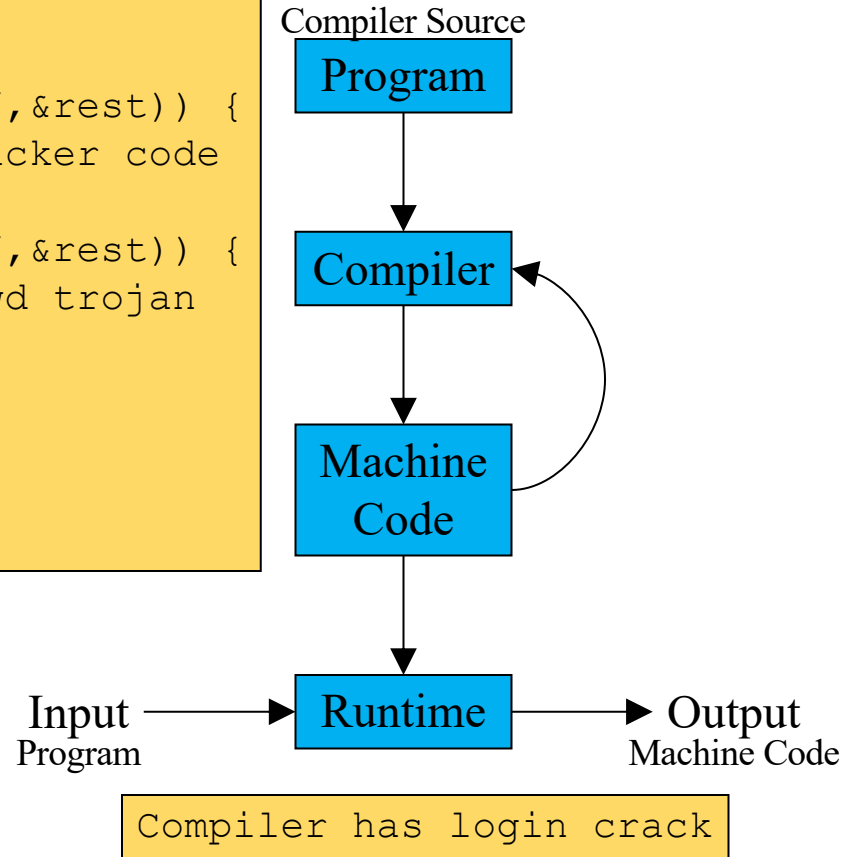
Get password
from system

Check entered
password against
system password

```
compile(char *s)
{
    if(match(s,"login",&rest)) {
        // add root passwd trojan
        compile(rest);
    }
    ...
}
```



```
compile(char *s)
{
    if(match(s,"compile",&rest)) {
        // insert login cracker code
        compile("
        if(match(s,"login",&rest)) {
            // add root passwd trojan
            compile(rest);")
        }
        compile(rest);
        ...
    }
}
```




```
compile(char *s)
{
    // standard compiler code
    // no login crack
    ...
}
```

[Reflections on Trusting Trust](#),
Ken Thompson.
CACM 27(8), pp. 761-763,
1984.

