CS342: Organization of Prog. Languages

Topic 19:
And Now For Something Completely Different

• John Backus, in memoriam.
• How well do you know C?
• Scheme 5 vs Common Lisp vs Scheme 6
• Fortran and Fortress

John Backus

- Studied mathematics at Columbia. Masters at age 25.
- 1950 joined IBM. Worked on project to calculate position of moon.
- 1954 assembled team to develop Fortran. First HLL to be used widely.
- Part of committee to develop Algol 58 and Algol 60. Invented BNF for this (notation for context-free grammars).
• Turing award citation:  
  *For profound, influential, and lasting contributions to the design of practical high-level programming systems, notably through his work on FORTRAN, and for seminal publication of formal procedures for the specification of programming languages.*

• 2007 died age 82.
How Well Do you Know C?

- The International Obfuscated C Code Contest
  http://www0.us.ioccc.org/main.html
- 19th IOCCC Competition closed Feb 28.
- Goals:
  - To write the most obscure/obfuscated C program under the rules.
  - To show the importance of programming style, in an ironic way.
  - To stress C compilers with unusual code.
  - To illustrate some of the subtleties of the C language.
  - To provide a safe forum for poor C code. :-)
- Rules (excerpt):
  - Complete, original ANSI C program ≤ 4096 bytes, ≤ 2048 bytes
    excluding white space, etc.
  - Legal abuse of the rules is somewhat encouraged.
Example

```c
int i;main(){for(;i["]<i;++i){--i;}};read(’-’-’-’,i+++"hello\n, world!\n",’/’/’/’));}read(j,i,p){write(j/p+p,i---j,i/i);}  

• Prints "hello world!", anonymous 1984.
```
What?

int i;main(){for(;i["]<i;++i){--i;}};read(’’’’,i+++"hello world!\n","'/'/''');}read(j,i,p){write(j/p+p,i---j,i/i);}
What?

int i;

main() {
    for( ;
        i["i;++i]{--i;}"
    read("-\-\-\-,i+++"hello, world!\n",’/’/’/’)
        ) ;
}

read(j,i,p) { write(j/p+p, i---j, i/i); }
What?

```c
int i;

main() {
    for( ; i["]<i;++i){--i;" ];
        read(0,i+++"hello, world!\n",'/'/'/');
    }
}

read(j,i,p) { write(j/p+p, i---j, i/i); }
```
What?

int i;

main() {
    for( ;
        i["]<i;++i){--i;}}
    read(0,i+++"hello, world!\n",1)
    );
}

read(j,i,p) { write(j/p+p, i---j, i/i); }
int i;

main() {
    for( ;
        i["<"]<i;++i){--i;}
    w(0,i+++"hello, world!\n",1)
    );
}

w(j,i,p) { write(0/1+1, i---0, i/i); }
int i;

main() {
    for( ; i["<i;+i"]{--i;}"];
        w(i+++"hello, world!\n")
    ) ;
}

w(i) { write(1, i--, 1); }
What?

int i;
char *t = "}]<i;++i){--i;}";
char *h = "hello, world!\n";

main() {
    for ( ;
        i[t];
        w(i+++h)
    );
}

w(i) { write(1, i--, 1); }
What?

int i;
char *t = "}]<i;++){--i;}";
char *h = "hello, world!\n";

main() {
    for( ; t[i] != 0; w(i+++h))
        ;
}

w(i) { write(1, i--, 1); }
What?

```c
int i;
char *t = "xxxxxxxxxxxxxxxxx";
char *h = "hello, world!\n";

main() {
    for( ; t[i] != 0; w(i+++h))
        ;
}

w(s) { write(1, s, 1); }
What?

```c
int i;
char *t = "xxxxxxxxxxxxxxxxxx";
char *h = "hello, world!\n";

main() {
    for( ; t[i] != 0; w(h+i), i++)
        ;
}

w(s) { write(1, s, 1); }
```
int i;
char *t = "xxxxxxxxxxxxxxxxxx";
char *h = "hello, world!\n";

main() {
    for( ; t[i] != 0; i++)
        w(h+i);
}

w(s) { write(1, s, 1); }
What?

int i;
char *h = "hello, world!\n";

main() {
    for(i = 0 ; i < 14; i++)
        w(h+i);
}

w(s) { write(1, s, 1); }
Other Examples

- Arachnid.c: an amazing program
  http://www0.us.ioccc.org/2004/arachnid.c
  http://www0.us.ioccc.org/2004/arachnid.hint

- Gavare.c: no keywords
  http://www0.us.ioccc.org/2004/gavare.c
  http://www0.us.ioccc.org/2004/gavare.hint

- Gavin.c: an operating system
  http://www0.us.ioccc.org/2004/gavin.c
  http://www0.us.ioccc.org/2004/gavin.hint

- Jason.c: an adventure game (think Zork)
  http://www0.us.ioccc.org/2001/jason.c
  http://www0.us.ioccc.org/2001/jason.hint

- Primenum.c: Not a prime number generator
  http://www0.us.ioccc.org/2000/primenum.c
  http://www0.us.ioccc.org/2000/primenum.hint

- Vic2.c: A prime number generator
  http://www0.us.ioccc.org/2004/vik2.c
  http://www0.us.ioccc.org/2004/vik2.hint
Scheme vs Common Lisp

- Scheme: Developed — 1970s by Gerald Jay Sussman and Guy Steele.
  Common Lisp: Developed — 1980s by committee.

- Scheme: Philosophy — minimalist
  Common Lisp: Philosophy — comprehensive.

- Scheme: Definition — 50 pages.
  Common Lisp: Definition — 1029 pages.

- Scheme: Namespaces — 1.
  Common Lisp: Namespaces — \( \geq 3 \).

- Scheme: Binding — lexical.
  Common Lisp: Binding — lexical or dynamic.

- ...

- Scheme v6: Designed by committee.
Common Lisp

- Attempt to provide what was actually used in large lisp applications in various dialects.

- Package model (imports, exports, namespaces, inheritance).
- Keyword arguments to functions
- setf
- defstruct
- complex arrays
- Separate function and value spaces.
- Common Lisp Object System (CLOS)
Common Lisp Applications

- Orbitz travel booking.
- Xanalys investigation sw (police, security, fraud prevention).
- Jak and Daxter playstation games.
- Maxima (open source version of Macsyma)
Fortran

- Fortran 1960..present
- Efficient numerical programming.
- Most serious scientific computation on supercomputers.
Fortran Overhang

• fixed format source
• Hollerith format
• spaces irrelevant
• array layout and aliasing
• call by reference
Fortran Merits

- Compilers vectorize loop computations.
- Primitive operations for smart inter-processor communication of array data (e.g. cshift)
- Data placement aware. Important in super computing.
Fortran Successors

- Several efforts to produce a "successor" to Fortran.
- F — Subset of Fortran 95
- Fortran 2003. OO, polymorphism, etc etc etc.
- Fortress at Sun,