Lexical Elements, Operators, and the C Cystem

C overview recap

- functions
  - structured programming
  - return value is typed
  - arguments(parameters)

```
float minimum(float x, float y)
{
    if (x < y)
        return x;
    else
        return y;
}
```

- pointers

```
char  *p,  s[100],  **twod;
```

- files

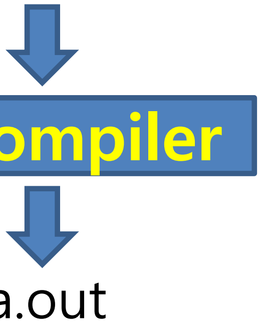
```
ifp = fopen(“my file”, “r”);
```

Syntax and Compiler

```
#include <stdio.h>

int main(void)
{
   int   i = 1, sum = 0;

   while (i <= 5) {
      sum += i;
      ++i;
   }
   printf("sum = %d₩n", sum);
   return 0;
}
```



C compiler

a.out

- The compiler understands some words and a simple grammar
- words
  - 32 keywords: if, int, return, ...
  - identifier; i, sum
  - constants: 1, 0, 5
  - string: “sum = %d₩n”
  - operators: =, <=, +=, ++
- grammar
  - keyword identifier ;
  - while (expression) statement | statements

Compiler

- Task
  - generate a binary file from a source file

1. check if all the words are correct (lexical analyzer)
2. check if the grammar is correct (syntax analyzer)

3. optimization
4. code generation

Identifiers

-
  ```
  <identifier> ::=
           <letter>
        |  <identifier> <letter>
        |  <identifier> <digit>
  ```
  - underscore is a letter in C
- some identifiers are used in the library
  - printf, scnaf, fopen, sqrt, pow, exp, sin, ..
  - _print

Constants

- integer
  - 17 is a decimal
  - 017 is an octal
  - 0x1F is a hexadecimal
- floating
  - 1.0
  - 3.14
- character ‘a’, ‘7’ ‘+’, ‘₩n’

String Constant

- “any thing you can put here”
- special chatacter should be preceded by a backslash
- it’s an array of characters
- library functions for strings
  - strcat, strcmp, strcpy, strleng

Operators

- Although C has very few keywords, it has a large number of operators

```
( )  [ ]  ->
!  ~  ++  --  +  -  * / %   &  (type)  sizeof
*  /  %
+  -
<<  >>
<  <=  >  >=
==  !=
&  ^  |
&&  ||
k = (n > 0) ? a : b;
=  +=  ....
```

Precedence

- 1 + 2 – 3 + 4 – 5   /* left to right */
- 1 + 2 * 3
- (1 + 2) * 3
- j *= k = m + 5   /* right to left */
- k++--   /* left to right */
- ++--k   /* right to left */

Increment and Decrement

- i++ and ++i are different

- Exercises

```
int  a=1, b=2, c=3, d=4;

a*b/c
a*b%c+1
++a  *  b  -  c—
7  -  -b * ++d
```

Strange Assignments

```
a = (b = 2) + (c =3);
a = b = c = 0;

j *= k + 3;
            /*  j = j*(k+3)  OR i = i*k +3  */

j *= k = m + 5;
```

```
/* Some powers of 2 are printed. */

#include <stdio.h>

int main(void)
{
    int    i = 0, power = 1;

    while (++i <= 10)
        printf("%6d", power *= 2);
    printf("₩n");
    return 0;
}
```

C Environments

- header files
  - usually contain only function prototype, NOT function code
  - then, where are the codes?
  - standard library is linked automatically
  - you should specify other libraries (math)

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int    i, n;

    printf("₩n%s₩n%s",
        "Some randomly distributed integers will be printed.",
        "How many do you want to see?  ");
    scanf("%d", &n);
    for (i = 0; i < n; ++i) {
        if (i % 10 == 0)
            putchar('₩n');
        printf("%7d", rand());
    }
    printf("₩n₩n");
    return 0;
}
```

gcc compiler

- gcc OR g++ are popular
  - C++ is a superset of C
- they call
  1. preprocessor
  2. compiler
  3. assembler
  4. linker

The **GNU Project** is a free software, mass collaboration project, announced on September 27, 1983, by Richard Stallman at MIT - wikipedia

gcc options

- perform parts of 4 steps
- specify output files and format; “gcc xxx.c –o x”
- -ansi OR specify differences
- warning options “ –w”
- debugging options “ –g “ “-ggdb” “-g3
- optimization “-O0” “-O2”
- preprocessor options
- assembler options
- linker options “-lm”
- ……