## Programming in C

- Program Environment
- Basic staff of C
- Memory management
- Pointer and Array
- Typedef Structure and Union
- Libraries





## Program Environment

- Available Compiler for C
  - gcc version 2.95.2 19991024 (release)    gcc -v
  - cc
- Debugger
  - gdb
  - ddd    GUI interface for gdb
  - purify
- Source file organisation





## File types

- Header file (.h)    Defination of data structures
- Source file (.c)    Declaration of data variable and code
- Object file (.o)    object binary code





## File Organisation

- Multiple Source Files and header file
- #include <   > and #include ””
- make
- gmake    (info make)

## Example Code of C

```
#include <stdio.h>
#include "myfile.h"
#define PI 3.1415926
main()
{
  float pi=3.1415926; int i=0, j=2;
  int month[12]={ 31,28,31,30,31,30,31,31,30,31,30,31 };
  printf("pi = %f\n", pi);
  { float pi = 3.14;
    printf("pi = %f\n", pi);
    printf("PI = %f\n", PI);
  }
}
```





## Compilation Stages

First the Preprocessor strips out comments and interprets directives (lines with a '#' character in the first column). The '#include' directive read in the named file, Further preprocessor directives to do with macros. If you want to just stop here, use "gcc -E filename"

After the preprocessor comes the compiler, which after 1 or 2 pass produces assembler code. If you want to get the assemble code, use "gcc -S filename"

Then the assemble compiler will turn the assemble code to machine object code. With "gcc -c filename" came to here.

Finally the link-loader produces the executable from the object file and any other object you mention.





## Debugging Procedure

- Compiling with '-g' flag
- ddd, gdb
- purify for memory leak
- errno for the error of system calls
- assert() putting diagnostics into programs





## Basic Types and Statements

- Void type: void
- Integral types: (signed/unsigned) char, short int, int, long, long long
- Floating-point types: float, double, long double
- No Boolean type for C

## Array Types and Constants

Array Consists of elements of same type

Examples:
int pages[10]; char line[80]; float index[Days][Month][Year];

**Note: the index start with 0!**

Char Constants: 'a', ' 0' Integer Constants: Can be decimal, octal,or hexadecimal. Append L for long, U for unsigned

Examples: 12L, 2000U

Floating-Point Constants: indicated by decimal point and/or exponent





## Initializers

You can initialize varibles when you define them.

float pi=3.1415926;

int i=0, j=2;

int month[12]={ 31,28,31,30,31,30,31,31,30,31,30,31 };

char greeting[25] = "Hello!";

char error[ ] = "Can not connect to remote server ";

This is initialize not assignment!





## Comments

Started with /* and ended with */

Can span lines, but can't be nested

Now you can also use //

Only work with one line

Examples:

/* Here is some comments */

int i; // I don't know why I define I here