

< >

: Al Kelly and Ira Pohl, "A Book on C", 4<sup>th</sup> Ed., Addison Wesley (1998)

: , " C " 가 (1991) 26000

Brian W. Kerningham and Dennis Ritchie, "The C Programming Language", 2<sup>nd</sup> Ed. Prentice Hall (1988)

: (40 %) + , (40 %) + (20 %)  
2 : , ,  
: 10 ( )  
: (50 % )

## 1. Introduction

### (1) Computing Environment

#### \* Classes of Computer

Microcomputer : PC / Desktop

Minicomputer : Engineering Workstation, Server

Mainframe, Supercomputer

#### \* Computing Environment

Server + Dummy Terminal Environment

Client – Server Environment

### (2) Software Environment

#### \* Component of Softwares

OS - UNIX, DOS, WINDOWS, ....

SERVER - MAIL SERVER, FILE SERVER, INTERNET SERVER

DBMS - ORACLE, SYBASE, SQL Server, My SQL, ...

DEVELOPMENT - C/FORTRAN/BASIC/PASCAL Compilers ,....

APPLICATIONS - Office, Game, Word Processor, ...

- \* Software Development

- Step.1 Algorithm Design

- Step.2 Program Composition

- Step.3 Debugging and Testing

- Step.4 Documentation

- Step.5 Storage and Maintenance

### Step.1 Algorithm Design

#### < Programming Style >

- \* virtue of good programmers in earlier times

- utilizes small amount of resource (memory, disk , ..)

- executed rapidly

- how quickly it can be written

- \* virtue of good programmers today

- reducing the cost of maintenance

- clarity and readability

- cooperation

#### < Modular Design, Top-Down Design, Structured Programming >

- \* Modular Design

- Divide programs in smaller subprograms (module)

- Benefits

- Easy to write and understand

- Can be debugged and tested easily

- Reusable

- Maintenance and modification is easy

- \* Top-Down Design

- Systematic development process that begins with general statement of program's objective and successively divides into detailed segments.

- “Divide and conquer”

- Benefit : less likely to overlook important operation

\* Structured Programming

Set of rules for good programming style

- clarity and readability

\* Algorithm Design

Flowcharting - using software (VISIO,...) , using diagram

Step. 2 Program Composition

\* Languages

FORTRAN (Formula Translation , 1957)

Suitable for numerical calculation (complex, double precision)

Large library of subroutines

Not suitable for graphics and end-user environment

BASIC (Beginner's All purpose Symbolic Instruction Code,1960s)

Easy for beginners

Extended to RAD (Rapid-Appilcation Development Tools) for RDBMS and other applications

PASCAL

Structured powerful language

C/C++

...

\* Structured Programming

programs = sequence + selection + repetition

only one entrance and one exit

GOTO should be avoided

Indentation

### Step.3 Quality control

#### Error of Bug

Syntax Error

Run-Time Error

Logical Error

#### Debugging

print intermediate result

use modular approach to localize error

trace

use calculator

think like computer

#### Testing

module test

development test

whole system test

typical data

unusual but valid data

incorrect data to check error-handling capabilities

#### Step. 4 Documentation

signature  
name of variables and their role  
insert space, indentation, labelling  
structured programming style

#### Step.5 Storage and Maintenance

Backup and Reuse old modules

#### \*\* History of C Language and UNIX Operating System

(196x) MUTICS Project – multi user OS (Bell LAB. /MIT)

(196x) Ken Thompson : UNIX (using PDP-7 assembler) “Space Travel”  
B-Language

(197x) Denis Ritchie : C-Language  
UNIX written in C Language

(197x) Berkley UNIX (MIT) Berkley Software Distribution (BSD)  
XENIX (Microsoft)  
UNIX/32V (VAX)  
System V (AT&T)

LINUX , GNU Declatration , ....

\* UNIX C :  
- UNIX : OS written by C language  
- C compiler is native to UNIX system

\* C : C++ , JAVA, PHP , ....