系統程式
System Programming

施吉昇教授/鄭卜壬教授
臺灣大學資訊工程系

Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

Who am I?

- 施吉昇 **Chi-Sheng Shih**

  PhD, University of Illinois at Urbana-Champaign

- **Research Interests:**
  - Real-Time Scheduling Theory
  - Embedded System Software
  - Multi-dimension Resource Management
  - Hardware/Software Co-Design for SoC
  - Assisted Living
- **Lab:**
  - Embedded Systems and Wireless Networking Lab
  - http://newslab.csie.ntu.edu.tw
- **My website:** http://www.csie.ntu.edu.tw/~cshih

Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

Who am I?

- 鄭卜壬 **Pu-Jen Cheng**
  PhD, National Chiao Tung University

- **Research Interests:**
  - Information Retrieval
  - Web Knowledge Discovery
  - Multimedia Database
  - Machine Learning
  - Chinese Information Processing
- **Lab:**
  - Web Mining & Information Retrieval Lab
- **My website:**
  - http://www.csie.ntu.edu.tw/~pjcheng

Announcement

- **Not in this class or session?**
  - Fill an enrollment request form if you are not enrolled in this class or this session and want to.
  - Drop the form in R 436 before March 2nd, 2007.
  - No guarantee for enrollment.

Why Should You be Here?

Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

Frankly, I don't give a damn.

說實在的，我才不鳥呢

Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

How a (Software) System Succeeds?

Creativity

Performance

Usability

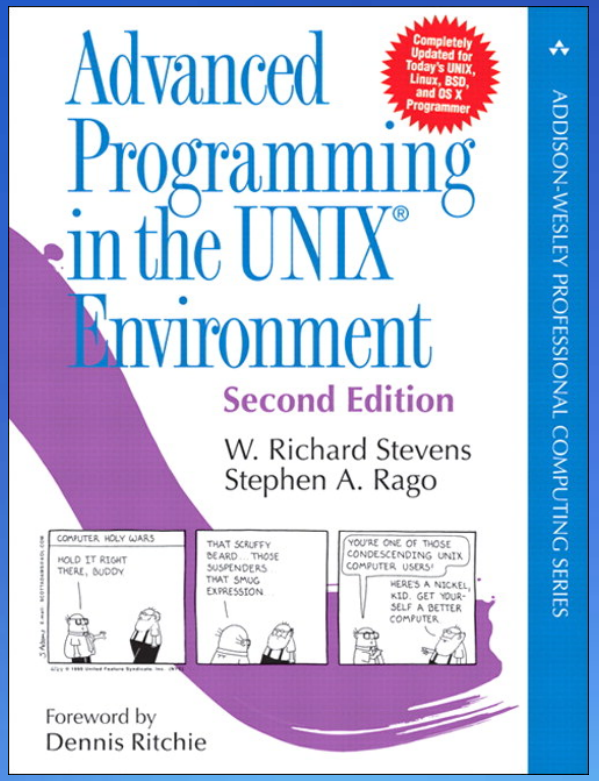
Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

Syllabus

- **Online Syllabus**
  - http://oris.csie.ntu.edu.tw/courses/SP2007
- **Take a syllabus**

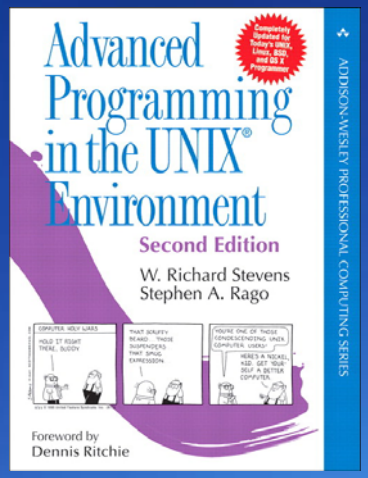
Textbook

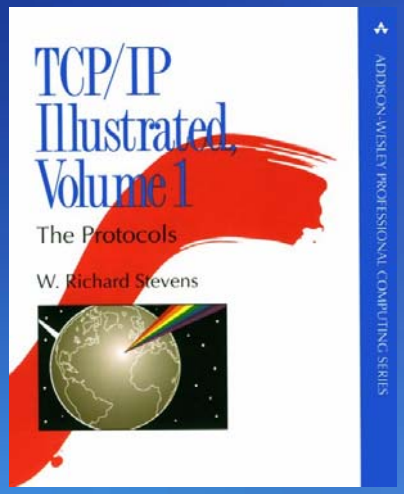
- **Required Text Book**
  - W. Richard Stevens and Steven A Rago, “Advanced Programming in the Unix Environment,” Addison-Wesley, 2nd Edition, 2005.
- **Reference Book:**
  - “Managing Projects With make” by Andrew Oram and Steve Talbott.
  - “The Art of UNIX Programming” by Eric S. Raymond.
    http://www.faqs.org/docs/artu/
- **Grading:**
  - Mid-term: 35%,
  - Final: 35%,
  - 4 programming assignments: 30%, and
  - weekly on-line exercise (for free).
- **Prerequisites:**
  - Data Structures, and C Programming.
- **Source codes shown in the text-book**
  - http://www.apuebook.com or
  - Our website

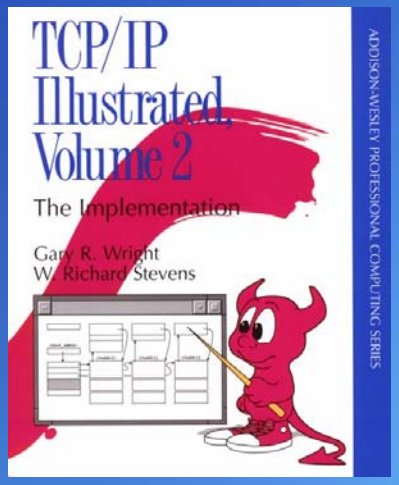


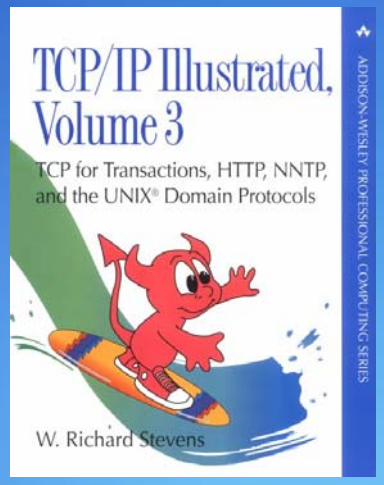
W. Richard Stevens (1951~1999)

Ph.D. (Systems Engineering), Univ. of Arizona, 1982
http://www.kohala.com/start/
http://en.wikipedia.org/wiki/W._Richard_Stevens









Administration Misc.

- **Web-site**
  - http://oris.csie.ntu.edu.tw/courses/SP2007
    - If you enrolled before Feb. 26, 2007, you already have an account on the site.
    - Using your student ID as your username and ‘SP2007’ as the password.
    - Please log in to change your password and your contact email.
    - Your account will be disabled if the password remains the same after March 12, 2007.
  - You will find the following on the site:
    - Program assignment announcement and hand-In.
    - Class slides
    - Forum
    - Free Exercise

The Most Important Skill

System Programming is an important course. However, learning how to learn effectively is the most important skills that you can acquire during college years:

- In engineering, new technologies and new knowledge are generated rapidly. Every 3-5 years, we need to significantly revise our course materials to reflect the changes.
- You must become an effective and efficient learner to get ahead.

How to learn Tightrope Walking?

- Buy a “Tightrope Walking 101” DVD
- Sit in the coach and watch DVD
- Then?

Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

Cone of Learning (Edgar Dale)

After 2 weeks, we tend to remember:

Passive learning

- 10% of what we read
- 20% of what we hear
- 30% of what we see (pictures)
- 50% of what we hear and see

Active learning

- 70% of what we say
- 90% of what we say and do

Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

Ways to Become an Active Learner

- Recall prior materials
- Answer a question
- Guess the solution first (even guessing wrong will help you to remember the right approach)
- Work out the next step before you have to read on
- Think of an application
- Imagine that you were the professor and think about how you would give a test on the subject material so that key concepts and results will be checked.
- Summarize a lecture, a set of home work or a lab in your own words concisely.

Are You Ready?

- **Now, you are a college student.**
- **What's the difference between college students and high school students?**
  - You are now an adult (by law.)
  - You will learn what you like to learn, not what the teacher asks to learn.
  - Who Give a Damn?
    - Bill Gates start his career when he was in college.
    - The first web browser are developed by several college students.
    - Linus Benedict Torvalds wrote his first Linux when he was in college.
    - …
- **You can do that too.**

UNIX

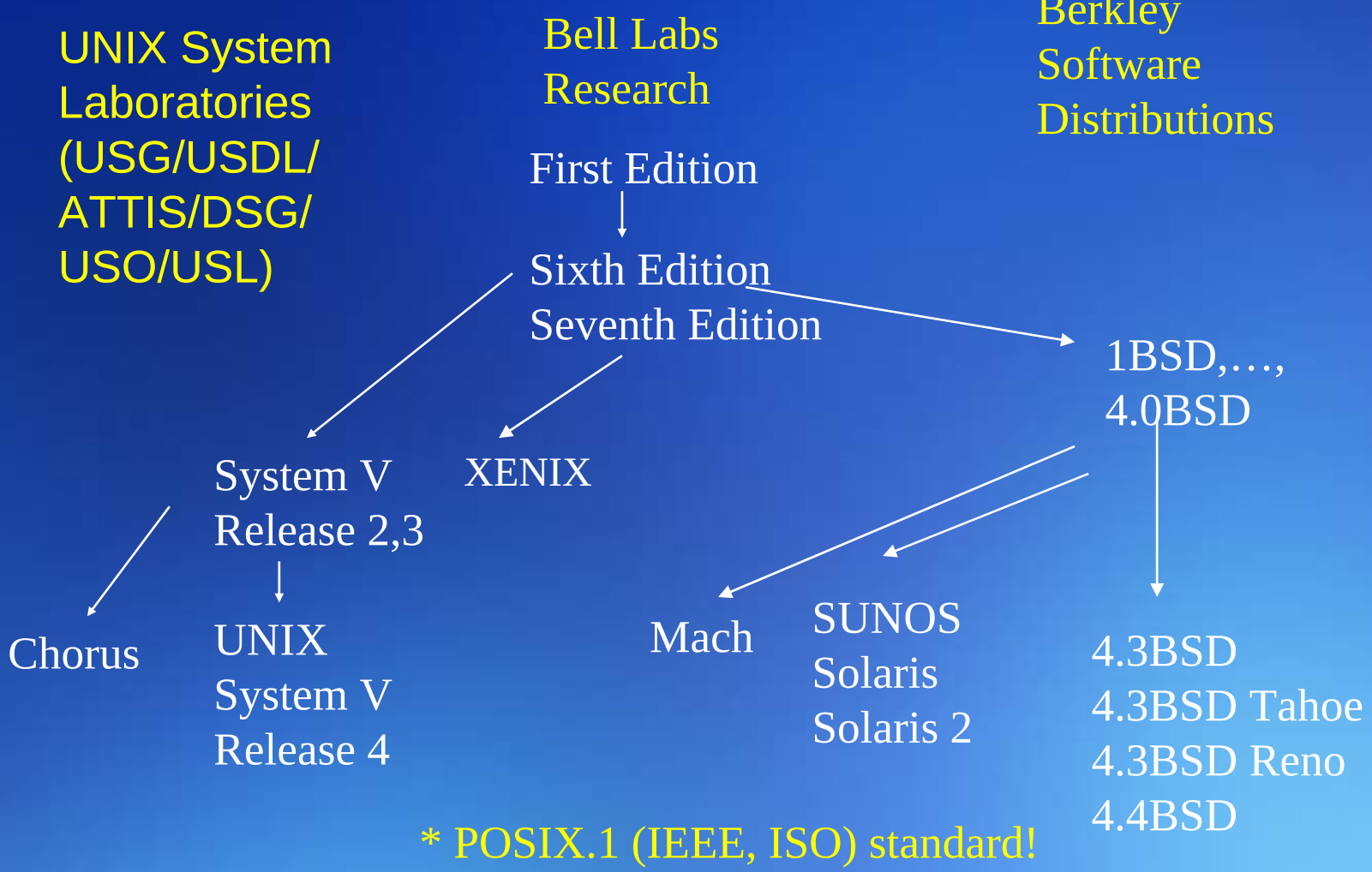
- **Created by Ken Thompson & Dennis Ritchie at Bell Laboratories in 1969 & on PDP-7.**
  - ACM Turing award winners for the design of UNIX in 1983.
  - C programming language inventor: Dennis Ritchie.
- **Major Contributors:**
  - Bell Laboratories, Computer Systems Research Group (CSRG) of the University of California at Berkley (released in BSD), UNIX System Laboratories (USG/USDL/ATTIS/DSG/USO/USL), etc.

PDP－7

PDP－11 (1972)
Ken (sitting) & Dennis (standing)

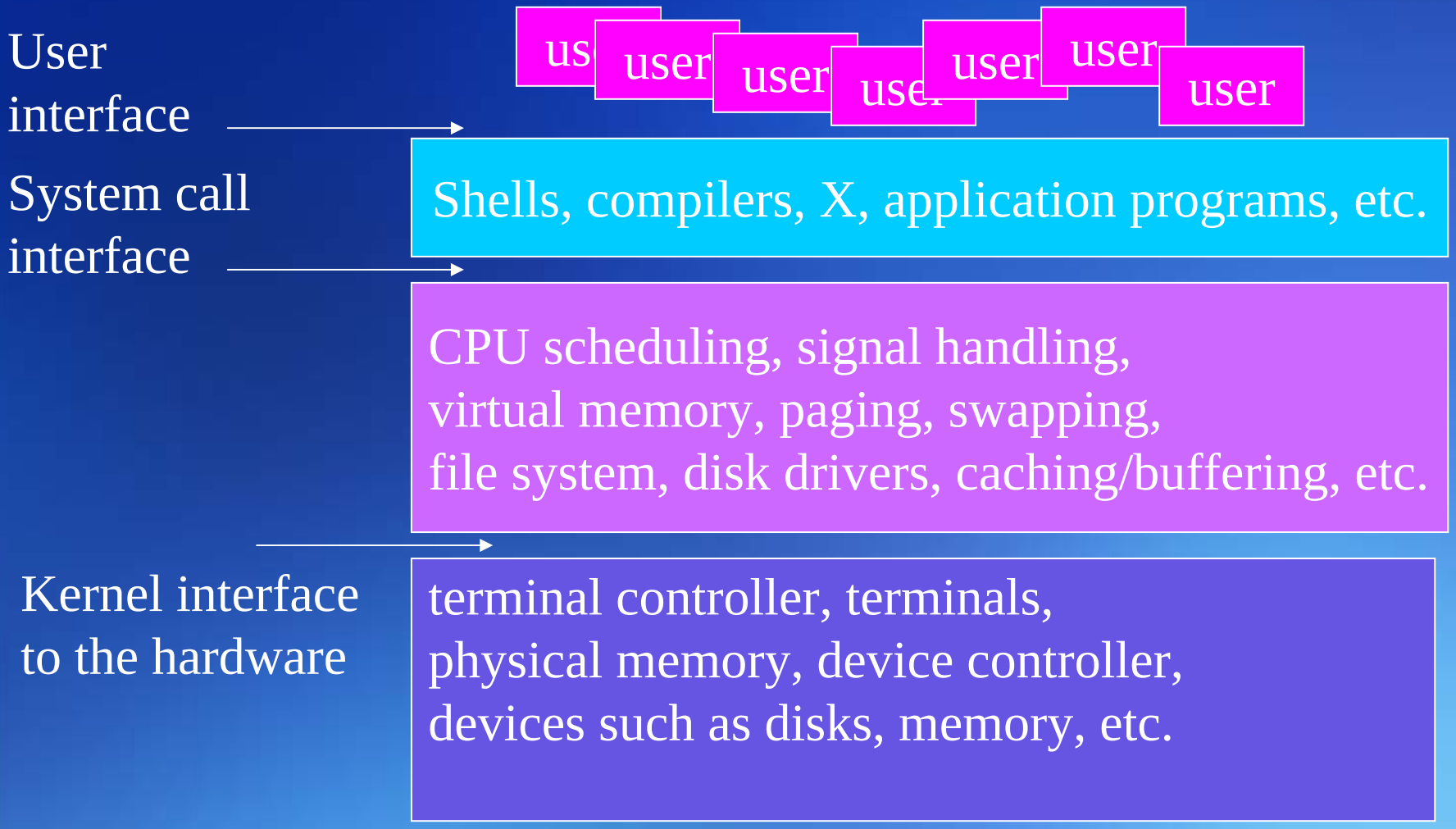
Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

UNIX



* POSIX.1 (IEEE, ISO) standard!

UNIX Architecture



UNIX

Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

Standards/Implementations

- **Related Implementations**
  - 4.xBSD, SVRx (ATT), SunOS
    - 4.3+BSD
  - uname -s
- **Standards**
  - POSIX.1
    - IEEE and ISO standards for the interface of Unix-like systems
  - ANSI C
  - XPG3 – X/Open Portability Guide, Issue 3

UNIX

- **Influence**
  - fork() from Berkley's GINIE, 4.2 BSD file-mapping virtual memory interface from TENEX/TOPS-20, 4.4BSD virtual memory interface from MACH. fcntl system call from System V. Disk quotas and 4.3 BSD time-zone-handling package from the user community.
  - 4BSD job control, reliable signals, multiple file-access permission groups, and file system interface were adopted by AT&T UNIX System V, IEEE POSIX.1 standard, etc. 4BSD socket ported to AT&T System III. 4BSD implementation of TCP/IP networking protocol suite widely adopted!

UNIX

- **Distinguishing Features**
  - Written nearly completely in a high-level language, i.e., C.
    - High portability!
  - Distributed in source form.
    - Contributions and bug fixing from everywhere!
  - Provide powerful primitives and functions such as concurrent processes.

Design Principles

- **Simple Algorithms for Implementation**
- **Replaceable Standard User Interface**
  - Shell
- **Time-Sharing**
  - Simple Priority-Driven CPU Scheduling
- **Demand-Paging Virtual Memory (4.3BSD)**
  - Swapping
- **Similar treatments of disk files and I/O devices**

Contents

- Preface/Introduction
- Standardization and Implementation
- File I/O
- Standard I/O Library
- Files and Directories
- System Data Files and Information
- Environment of a Unix Process
- Process Control
- Signals
- Inter-process Communication
- Thread Programming

Assignment 0

- **Read Chapter 1 and 2 in your textbook. In the next class, we will discuss**
  - Is Linux a UNIX system? How about Solaris and OS X?
  - Why did Ariane 5 self-destruct in less than one minute after its first launch on June 4, 1996?
- **Write a C program which reads a number of words from a file in argv[1], look for words with an initial 'A', sort them in the alphabetic order, and prints sorted words out at stdout.**
  - We will ask volunteers to show
    - His/her codes,
    - Makefile, and
    - How he/she compiled his/her codes.
- **Explain "variable-length argument lists"**
  - void printf(char *fmt, …)
  - va_start(ap, fmt), va_end(ap), va_arg(ap, int), va_list, <stdarg.h>
- **Due on March 13rd, 2007.**

ENJOY AND HAVE FUNS!!

Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

How to write your first C program in Linux

- **Preparation:**
  - Linux workstation account or your own copy
  - Compiler: gcc
  - Editor: vi/vim, emacs, or many others.
  - Debugger: gdb and ddd.

Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University

Using Your Linux Workstation Account – Windows Clients

- **Linux workstations now only accept secure remote connection:**
  - ssh: www.ssh.com
  - puTTY:
- **If you don't like command-lines environment,**
  - X-Win
  - Cygwin and xwindows components.

A sample program: sort integers

sample.c

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

static void shell_sort(int a[], int size)
{
        int i, j;
        int h=1;

        do {
                h = h * 3 + 1;
        } while (h <= size);
        do {
                h /= 3;
                for (i = h; i < size ; i++)
                        {
                                int v=a[i];
                                for (j = i; j >=h && a[j-h]> v; j-=h)
                                        a[j] = a[j-h];
                                if (i != j)
                                        a[j] = v;
                        }
        } while (h != 1);
}
```

```
int main( int argc, char *argv[])
{
        int *a;
        int i;

        a = (int *)malloc((argc - 1) * sizeof(int));
        for (i = 0; i < argc - 1; i++)
                a[i]=atoi(argv[i+1]);

        shell_sort(a, argc);

        printf("Output:");
        for (i=0; i < argc-1; i++)
                printf("%d ", a[i]);

        printf("\n");

        free(a);

        return 0;
}
```

Compile and execute the program

- >gcc –g –c sample.c
- >gcc –g –o sample sample.o
- >sample 9 7 8: is it correct?
- >sample 11 14: is it correct?

Debug Your Program using DDD

- Set breakpoints at lines 31, 35, and 37 by pressing the right button of your mouse in that line at the left border of the source window and selecting Set Brackpoint from the resultant context menu.
- Run the program by selecting Program->Run from the menu.
- Insert your failure-producing arguments and select Run.
- Show the content for an array:
  - a[0]@2
  - argv[0]@3

Makefile

```
TOP=sample

all: $(TOP)

$(TOP): $(TOP).o
	gcc -g -o $(TOP) $(TOP).o

$(TOP).o: $(TOP).c
	gcc -g -c $(TOP).c

clean:
	rm -f $(TOP).o $(TOP)
```

Version Control Tool

- **Using version control tool to story your programs:**
  - CVS
  - SVN
- **Version control tool can help you to**
  - Track the changes of your program,
  - Roll back to your earlier version, and
  - Collaborate with others.

ENJOY AND HAVE FUNS!!

Tei-Wei Kuo and Chi-Sheng Shih©2007
Department of Computer Science and Information Engineering
Graduate institute of Multimedia and Networking, National Taiwan University