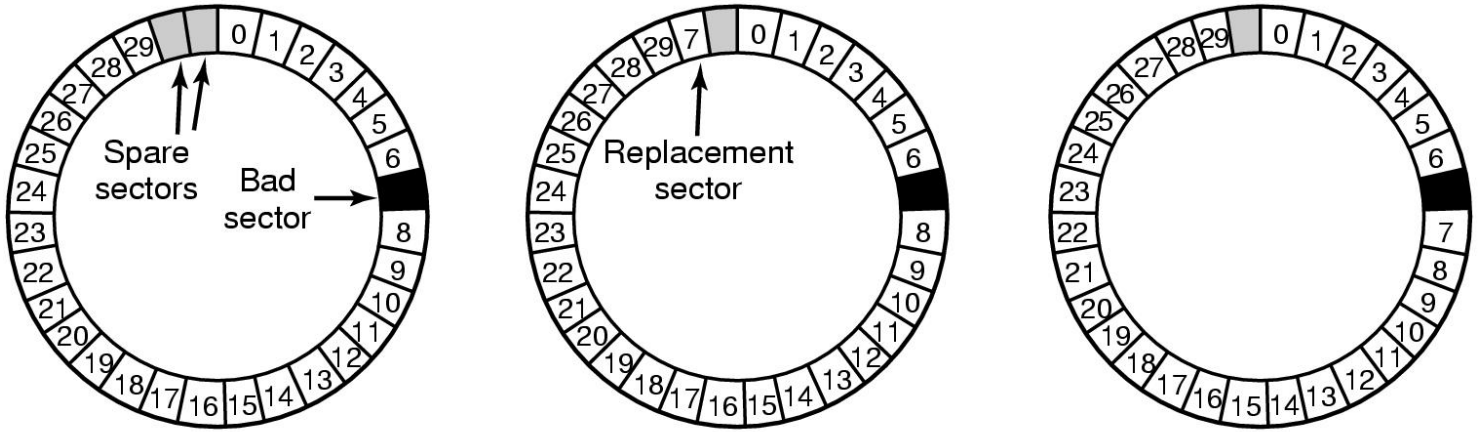
Operating Systems

Jinghui Zhong （钟竞辉）
Office： B3-515
Email： jinghuizhong@scut.edu.cn

Error Handling

Why do errors always happen?

As soon as manufacturing technology has improved to the point where it is possible to operate flawlessly at certain densities, disk designers will go to higher densities to increase the capacity.



Two solutions: Replace bad sector with spare sectors, or shift the sectors.

(a) A disk track with a bad sector;

(b) Substituting a spare for the bad sector;

(c) Shifting all the sectors to bypass the bad one.

Stable Storage

- RAIDs do not protect write errors laying down bad data in the first place.
- In some applications, it is essential that data never be lost or corrupted.
- Stable storage: the goal is to keep the disk consistent at all costs.

Stable Writes

① Write the block on drive 1, then read it to verify.

② If something wrong, write and reread again up to *n* times until they work.

③ After *n* consecutive failures, the block is remapped onto a spare and the operation repeated until it succeeds.

④ After the write to drive 1 has succeeded, the corresponding block on drive 2 is written and reread, repeatedly if need be, until it, too, finally succeeds.

Stable Read

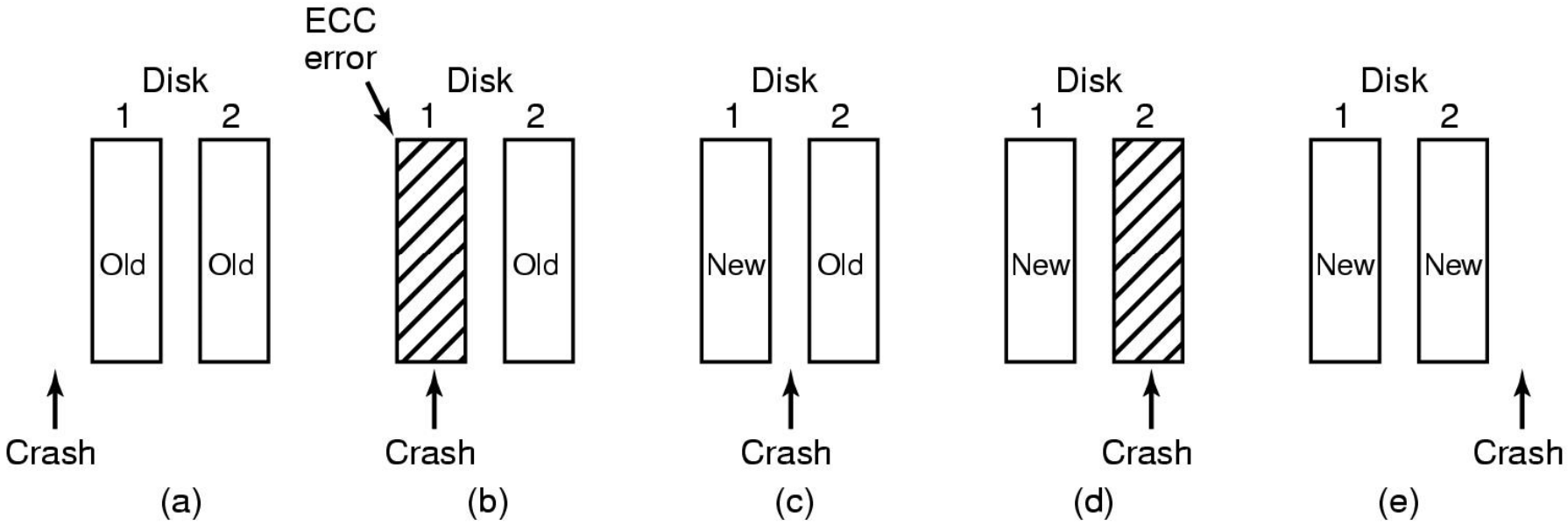
① Read the block on drive 1, if this yields an incorrect ECC, the read is tried again, up to *n* times.

② If all *n* times fail, then read from drive 2.

The probability of the same block going bad on both drivers is negligible.

Crash Recovery

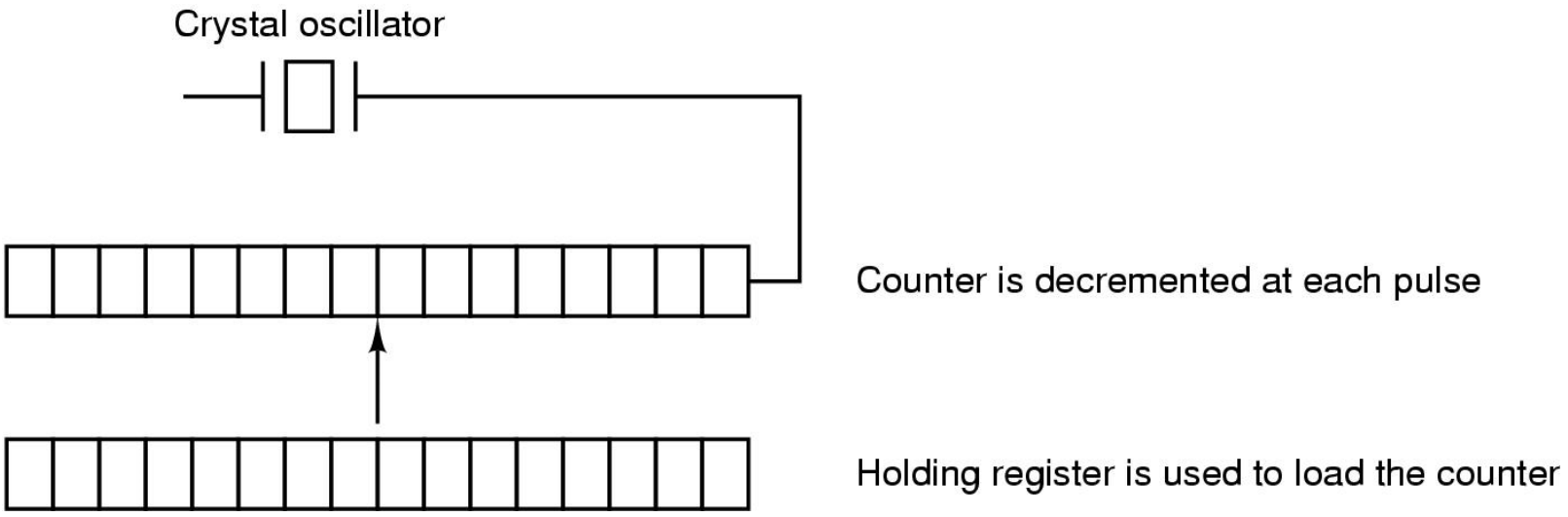
① If a pair of blocks are both good and the same, nothing is done;
② If one of them has an ECC error, the bad one is overwritten with the good one;
③ If a air blocks are both good but different, then block from drive 1 is written onto drive 2.



Analysis of the influence of crashes on stable writes

Clock Hardware

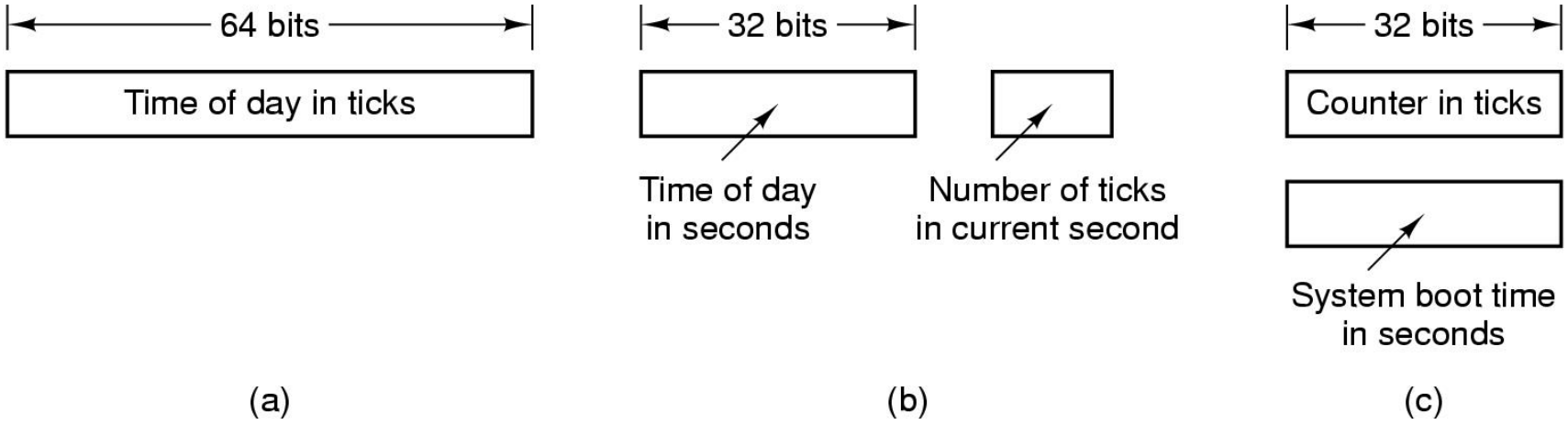
- The crystal oscillator can generate a periodic signal in the range of several hundred MHz.
- Two modes: one-shot mode, and square-wave mode.
- Clock ticks: periodic interrupts caused by the programmable clock.



A programmable clock.

Clock Software

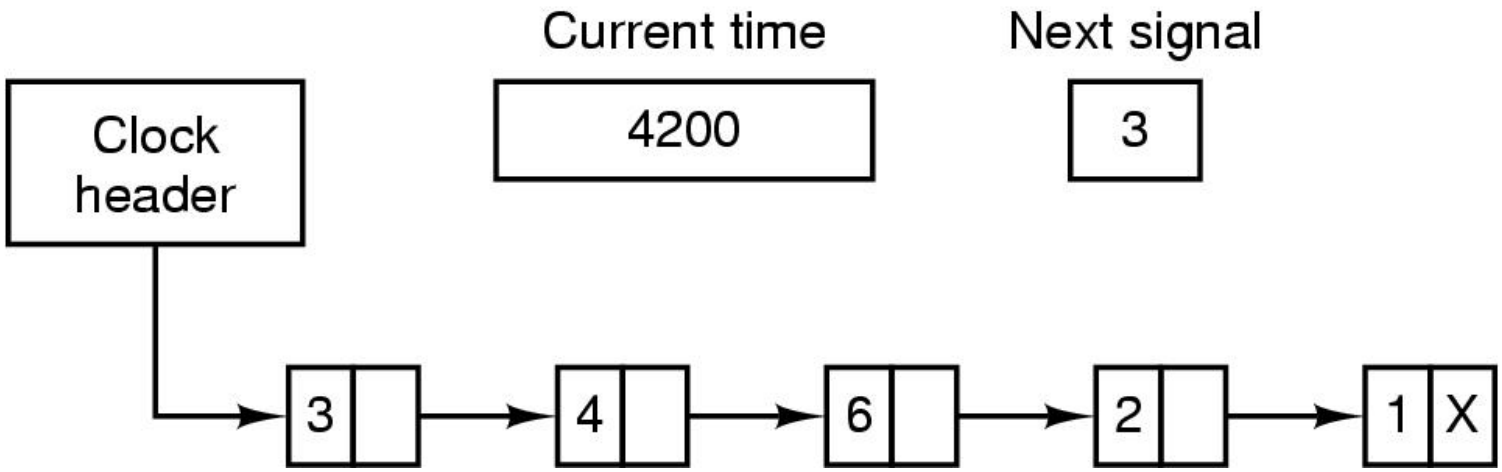
- The functions of clock driver
  - ✓ Maintaining the time of day.
  - ✓ Preventing processes from running too long.
  - ✓ Handling the alarm system calls (e.g., ACK).
  - ✓ Others.



Three ways to maintain the time of day.

Clock Software

- To satisfy multiple timing requests, the OS needs to simulate multiple virtual timers with a single clock.



Problems

① The clock interrupt handler on a certain computer requires 2 msec (including process switching overhead) per clock tick. The clock runs at 60 Hz. What fraction of the CPU is devoted to the clock?

② A computer uses a programmable clock in square-wave mode. If a 500MHz crystal is used, what should be the value of the holding register to achieve a clock resolution of

(a) a millisecond (a clock tick once every millisecond)?

(b) 100 microseconds?

Input Software: Keyboard

- An interrupt is generated when a key is pressed or released.
- The key board driver extracts the Scan code from the I/O port, and translates it to ASCII code.
- Two modes: canonical mode, Noncanonical mode.

```
GET_KEY: MOV AH,1
         INT   21H
         CMP AL,'Y'
         JE YES
         CMP AL,'N'
         JE NO
         JNE GET_KEY
```

Special Characters

Character	Posix Name	Comment
CTRL-H	ERASE	Backspace on character
CTRL-U	KILL	Erase entire line being typed
CTRL-V	LNEXT	Interpret next character literally
CTRL-S	STOP	Stop output
CTRL-Q	START	Start output
DEL	INTR	Interrupt process
CTRL-\	QUIT	Force core dump
CTRL-D	EOF	End of file
CTRL-M	CR	Carriage return
CTRL-J	NL	linefeed

Characters that are handled specially in canonical mode.

Mouse Software

- Two common types:
  - ✓Mouse with a rubber ball
  - ✓Optical mouse

- When a mouse has moved a certain minimum distance or a button is depressed or released, a message is sent to the computer.

  (Dx, Dy, status of buttons)

Output Software: Text Windows

Escape sequence	Meaning
ESC [*n*A	Move up *n* lines
ESC [*n*B	Move down *n* lines
ESC [*n*C	Move right *n* spaces
ESC [*n*D	Move left *n* spaces
ESC [*m*;*n*H	Move cursor to (*m*,*n*)
ESC [*s*J	Clear screen from cursor (0 to end, 1 from start, 2 all)
ESC [*s*K	Clear line from cursor (0 to end, 1 from start, 2 all)
ESC [*n*L	Insert *n* lines at cursor
ESC [*n*M	Delete *n* lines at cursor
ESC [*n*P	Delete *n* chars at cursor
ESC [*n*@	Insert *n* chars at cursor
ESC [*n*m	Enable rendition *n* (0=normal, 4=bold, 5=blinking, 7=reverse)
ESC M	Scroll the screen backward if the cursor is on the top line

The ANSI escape sequences accepted by the terminal driver on output. ESC denotes the ASCII escape character (0x1B), and, *n*, *m* and *s* are optional numeric parameters.

e.g., ESC[3;1H

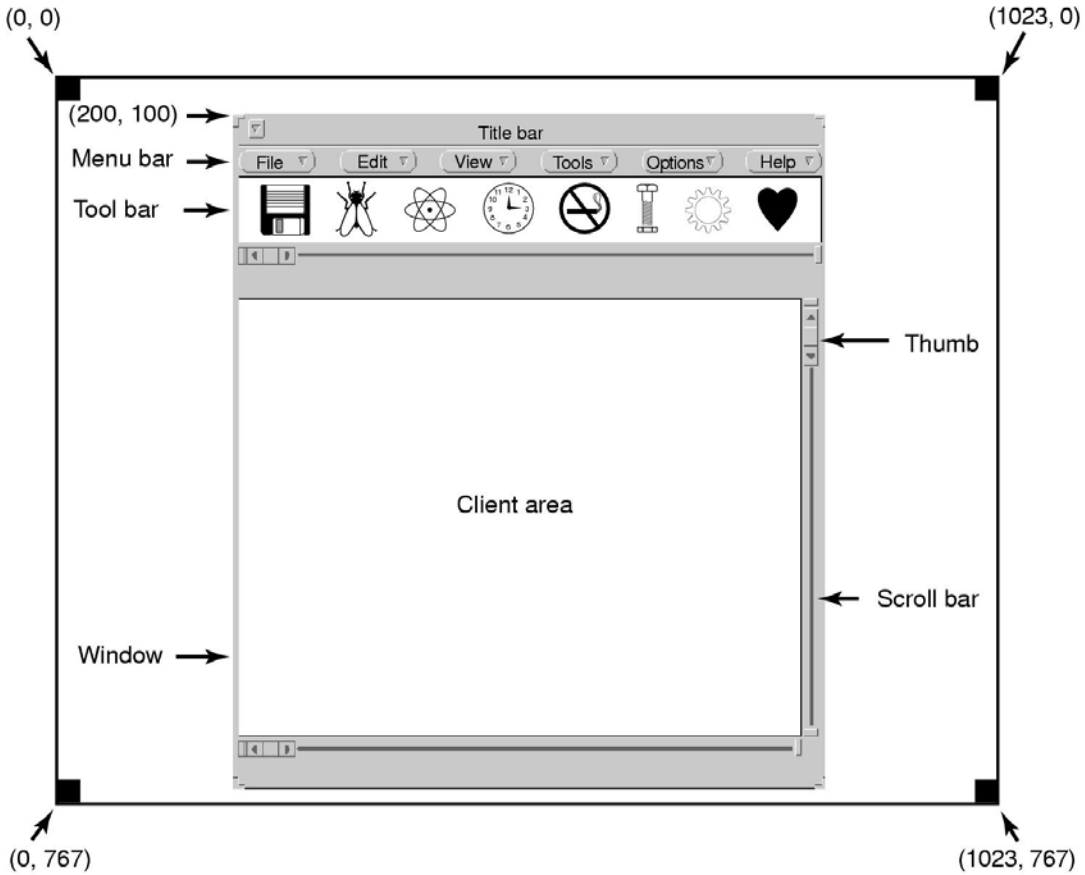
Problems

① The designers of a computer system expected that the mouse could be moved at a maximum rate of 20 cm/sec. If a mickey is 0.1 mm and each mouse message is 3 bytes, what is the maximum data rate of the mouse assuming that each mickey is reported separately?

② Assuming that it takes 2 nsec to copy a byte, how much time does it take to completely rewrite the screen of an 80 character * 25 line text mode memory-mapped screen? What about a 1024 * 768 pixel graphics screen with 24-bit color?

Output Software: GUI Windows

- A GUI (Graphical User Interface) has four essential elements: Windows, Icons, Menus, and Pointing device.



A sample window located at (200,100) on an XGA display.

Windows Output Software (2)

```
#include <windows.h>

int WINAPI WinMain(HINSTANCE h, HINSTANCE, hprev, char *szCmd, int iCmdShow)
{
      WNDCLASS wndclass;              /* class object for this window */
      MSG msg;                        /* incoming messages are stored here */
      HWND hwnd;                      /* handle (pointer) to the window object */

      /* Initialize wndclass */
      wndclass.lpfnWndProc = WndProc;          /* tells which procedure to call */
      wndclass.lpszClassName = "Program name";    /* Text for title bar */
      wndclass.hIcon = LoadIcon(NULL, IDI_APPLICATION);   /* load program icon */
      wndclass.hCursor = LoadCursor(NULL, IDC_ARROW);    /* load mouse cursor */

      RegisterClass(&wndclass);       /* tell Windows about wndclass */
      hwnd = CreateWindow ( ... )     /* allocate storage for the window */
      ShowWindow(hwnd, iCmdShow); /* display the window on the screen */
      UpdateWindow(hwnd);             /* tell the window to paint itself */
```

A skeleton of a Windows main program (part 1)

Windows Output Software (3)

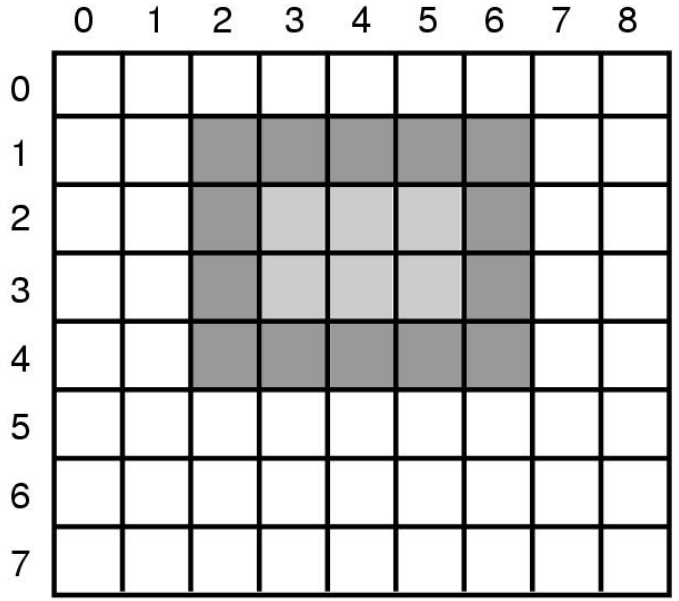
```
        while (GetMessage(&msg, NULL, 0, 0)) {    /* get message from queue */
                TranslateMessage(&msg);    /* translate the message */
                DispatchMessage(&msg);     /* send msg to the appropriate procedure */
        }
        return(msg.wParam);
}

long CALLBACK WndProc(HWND hwnd, UINT message, UINT wParam, long lParam)
{
        /* Declarations go here. */

        switch (message) {
                case WM_CREATE:     ... ;   return ... ;     /* create window */
                case WM_PAINT:      ... ;   return ... ;     /* repaint contents of window */
                case WM_DESTROY:    ... ;   return ... ;     /* destroy window */
        }
        return(DefWindowProc(hwnd, message, wParam, lParam));/* default */
}
```

A skeleton of a Windows main program (part 2)

Draw points, lines, rectangles

- The Win32 SDK contain many APIs for GUI development.

```
hdc=GetDC(hwnd);
Rectangle(hdc, xleft, ytop,
xright,ybottom);
MoveToEx(hdc,x, y,&tp);
LineTo(hdc,x, y);
SetPixel(hdc, x, y, color);
```

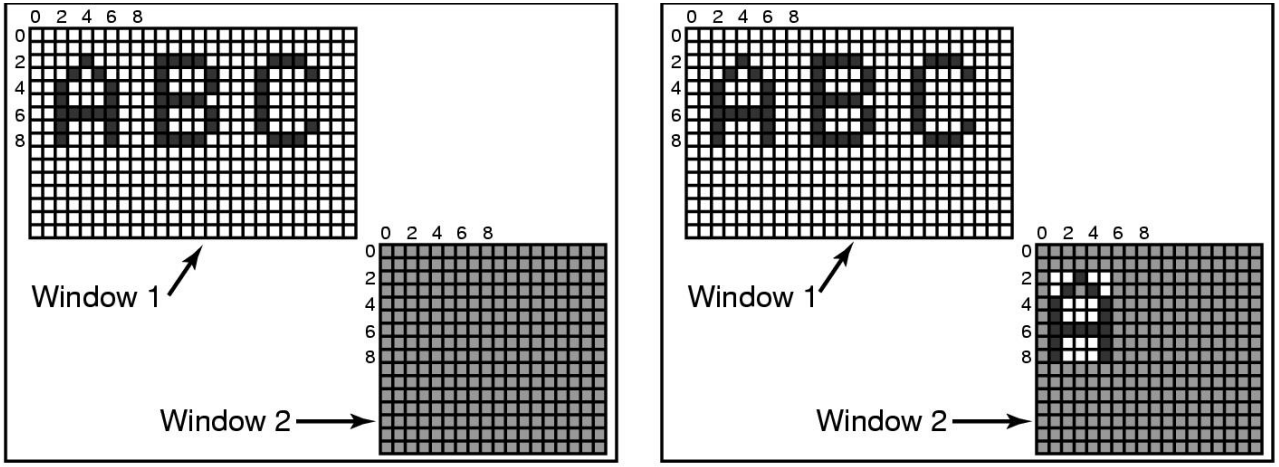


An example rectangle drawn using Rectangle, each box represents one pixel.

Bitmap

- Copying bitmaps using Bitblt.

```
BITMAP bm;
HDC hMemDC;
POINT pt;
hMemDC = ::CreateCompatibleDC(hdc);
::SelectObject(hMemDC,hBitmap);
::GetObject(hBitmap,sizeof(BITMAP),&bm);
pt.x = bm.bmWidth;
pt.y = bm.bmHeight;
::BitBlt(hdc,xStart,yStart,pt.x,pt.y,hMemDC,0,0,SRCCOPY);
::DeleteDC(hMemDC);
```



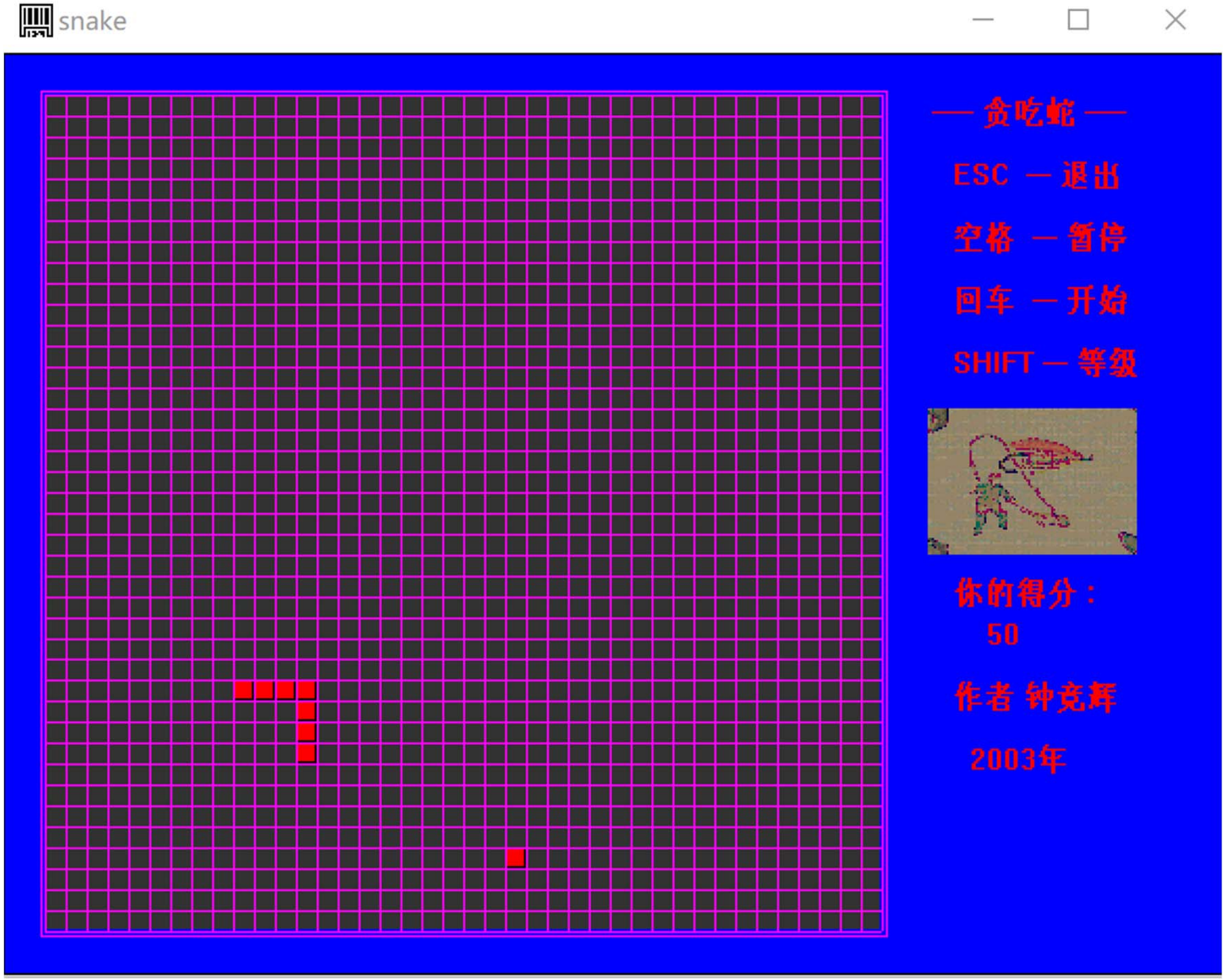
Fonts

- Each Truetype Font is defined by a sequence of points around its perimeter. All the points are relative to the (0,0) origin.

Some examples of character outlines at different point sizes.

```
hFont = CreateFont (
  20,0,     //高度20，宽取0表示由系统选择最佳值
  0, 0,     //文本倾斜，与字体倾斜都为0
  FW_HEAVY,     //粗体
  0,0,0,         //非斜体，无下划线，无中划线
  GB2312_CHARSET,     //字符集
  OUT_DEFAULT_PRECIS,
  CLIP_DEFAULT_PRECIS,
  DEFAULT_QUALITY,          //一系列的默认值
  DEFAULT_PITCH | FF_DONTCARE,
  "自定义字体"     //字体名称 );
SetTextColor(hdc, RGB(0, 255, 0));
SelectObject(hdc, hFont);
TextOut(hdc, x, y, lpszTx2, strlen(lpszTx2));
```

Snake: An example of Win GUI Programming



Power Management (1)

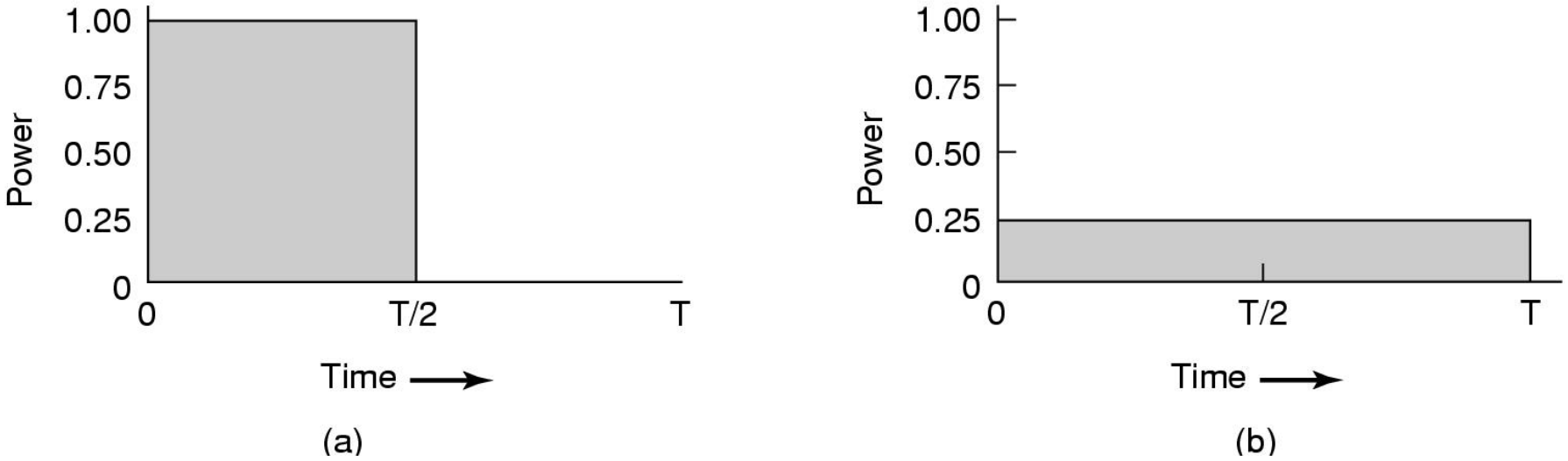
Device	Li et al. (1994)	Lorch and Smith (1998)
Display	68%	39%
CPU	12%	18%
Hard disk	20%	12%
Modem		6%
Sound		2%
Memory	0.5%	1%
Other		22%

Power consumption of various parts of a notebook computer.

The most common method to save battery is to design the devices to have multiple states:

On, Sleep, and Off.

Power Management (2)



(a) Running at full clock speed.

(b) Cutting voltage by two cuts clock speed by two and power consumption by four.

Power Management (3)

- The user can run longer on a given battery by accepting some quality degradation.
  - ✓Abandon the color information and display the video in black and white.
  - ✓Use radio link to send task to other devices.
  - ✓Trading image quality to reduce the transmission overload.

Problems

- If a CPU's maximum voltage, V, is cut to V/n, its power consumption drops to 1/n^2 of its original value and its clock speed drops to 1/n of its original value.
- Suppose that a user is typing at 1 char/sec, but the CPU time required to process each character is 100 msec.
- What is the optimal value of n and what is the corresponding energy saving in percent compared to not cutting the voltage?
- Assume that an idle CPU consumes no energy at all.

Summary

- Two kinds of I/O devices.
- The components of I/O device.
- Three ways to communicate with I/O.
- Disks: Magnetic disks (components, disk arm scheduling), RAIDs
- Clocks
- Terminals: Text windows, GUI Windows
- Power Management