System Software

AS Computer Science ▪ Topic 5

A-Level Computer Science

applications
operating system
hardware

What we will cover

1. Operating systems
2. Utility software
3. Program libraries
4. Language translators
5. IDEs
6. Recap

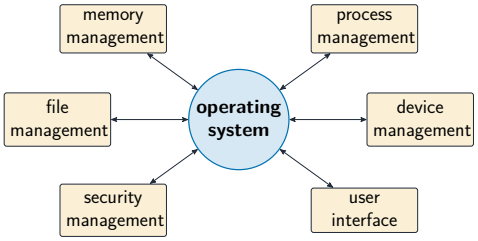
1 Operating systems

Why a computer needs an OS

The **operating system** 操作系统 is the software layer that:

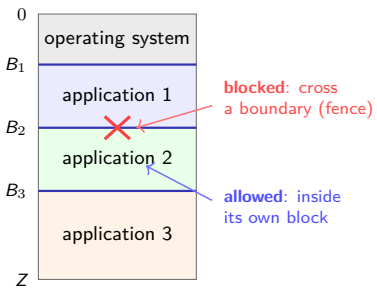
- manages the hardware for programs
- provides services & a user interface
- lets programs share the hardware safely

What the OS manages



- **process management** 进程管理 – schedule the CPU
- **memory management** 内存管理 – **virtual memory** 虚拟内存, paging
- file & **device management** 设备管理 (drivers)
- security, UI, networking

Memory protection and utilities



each app kept in its own block

Utility programs 实用程序 maintain the system:

- **antivirus** 杀毒软件, **firewall** 防火墙
- **backup** 备份, **compression** 压缩
- **disk defragmenter** 碎片整理 (HDD only)

Which utility would you use?

an old hard disk reads files slowly – pieces scattered everywhere

→ **disk defragmenter**

an attachment is too big to email

→ **compression**

one careless delete could lose the whole project

→ **backup**

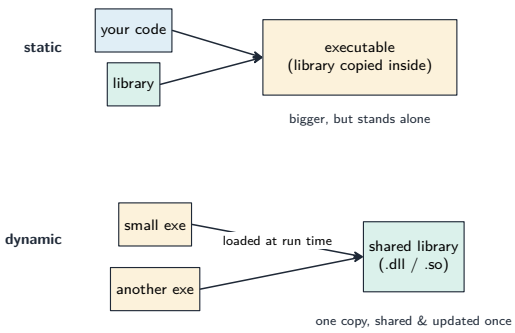
a borrowed USB stick might carry malware

→ **virus checker**

An exam answer names the utility and links it to the scenario's need.

2 Libraries

Program libraries

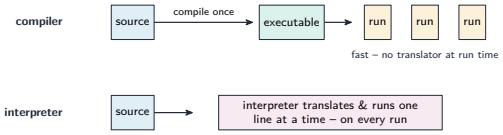


A **program library** 程序库 is reusable, tested code.

- **static library** 静态库: copied into the executable
- **dynamic library** 动态库 (`.dll`/`.so`): loaded at run time, shared

3 Translators

Compiler vs interpreter



- **compiler** 编译器: translates once → standalone **executable** 可执行文件; faster to run
- **interpreter** 解释器: line by line, every run; easy to debug & port

When to use which

use a compiler

run-time speed matters
ship to users without dev tools
runs many times

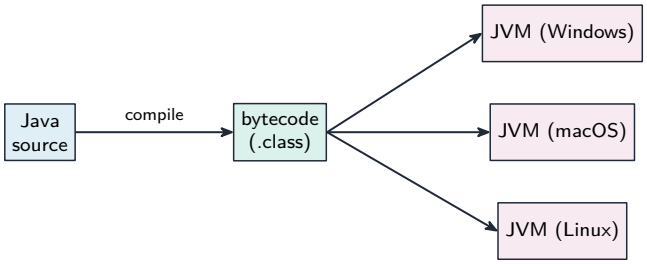
use an interpreter

fast edit–run cycles
cross-platform scripts
teaching beginners

Hybrid: Java bytecode

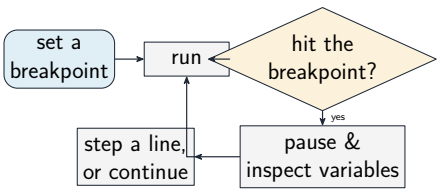
Java compiles to **bytecode** 字节码, run by a **virtual machine** 虚拟机 (JVM).

> Errors caught early, runs anywhere with a JVM – "write once, run anywhere";
> **JIT** 即时编译 reaches near-native speed.



4 IDE

Integrated Development Environment



An **IDE** 集成开发环境 bundles the dev tools:

- editor with **syntax highlighting** 语法高亮, **auto-complete** 自动补全
- one-key compile/run; inline errors
- **debugger** 调试器: **breakpoints** 断点, step, inspect

5 Recap

Topic 5 – key takeaways

1 OS

manages CPU, memory, files, devices, security

2 Libraries

reusable code; static vs dynamic

3 Translators

compiler once vs interpreter line-by-line

4 IDE

editor, translator & debugger in one

Check yourself

1. Which OS job keeps each program in its own memory?
2. Compiler or interpreter: which makes a standalone file?
3. Why is Java bytecode portable?
4. Name a debugger feature in an IDE.

Answers **1.** memory management (protection) **2.** compiler **3.** any JVM runs it **4.** breakpoints / step / inspect