Operating Systems Structures

DSC 315: Computer Organization & Operating Systems

Dr. Laltu Sardar

School of Data Science,
Indian Institute of Science Education and Research Thiruvananthapuram (IISER TVM)

March 08, 2026

1. Operating Systems Structures
   - Operating System Services
   - System Calls
   - API – System Call – OS Relationship
   - System Services
   - The Role of the Linker and Loader
   - Why Applications are Operating System Specific

Operating Systems Structures

Operating System Services

Environment for executing programs and supporting users

- **User Services**
  - **User Interface** – interaction with system (CLI, GUI, touch)
  - **Program Execution** – load, run, and terminate programs
  - **I/O Operations** – read and write to devices or files
  - **File-System Manipulation** – create, delete, read, write files
  - **Communication** – exchange data between processes
  - **Error Detection** – detect and handle system errors
- **System Services**
  - **Resource Allocation** – manage CPU, memory, and devices
  - **Logging** – record usage of system resources
  - **Protection and Security** – control access and authenticate users

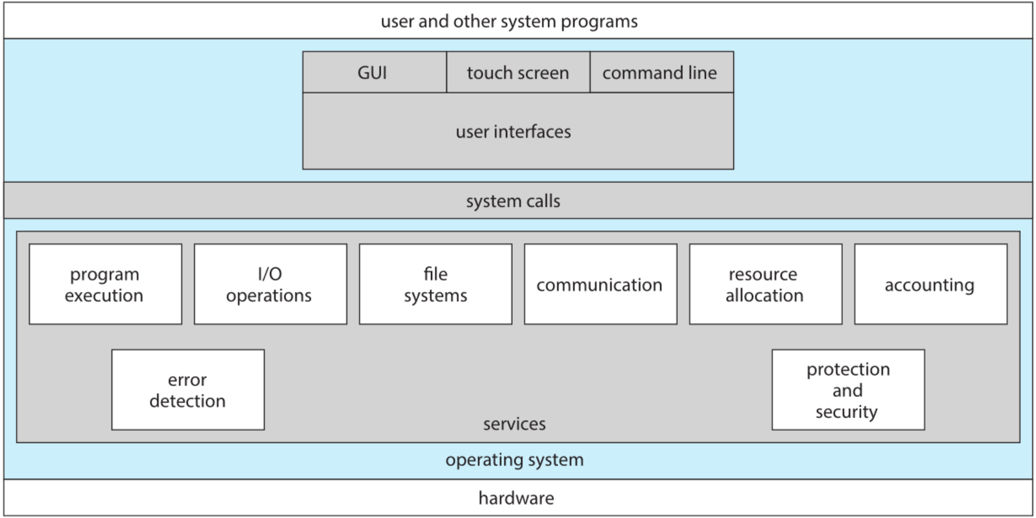
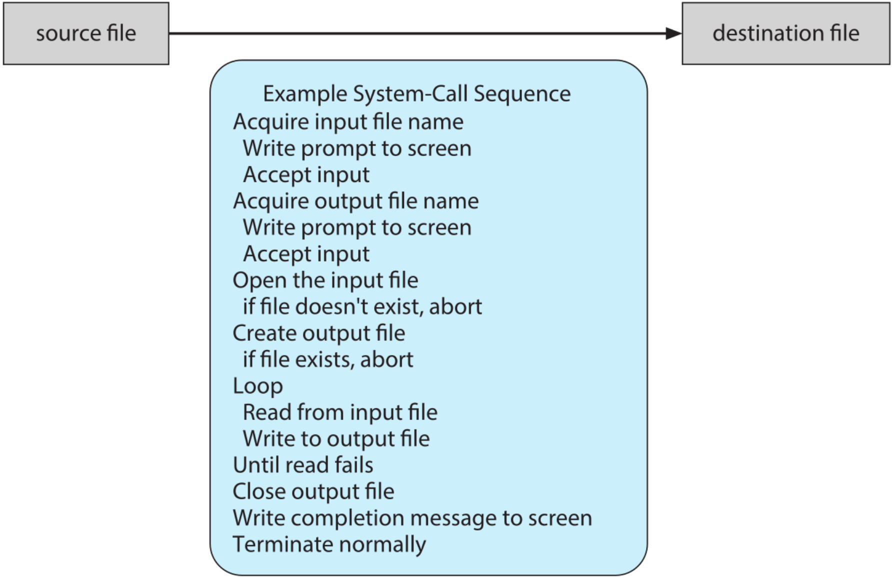


Figure: A view of operating system services.

System Calls

System Calls

- **System Calls –** programming interface to services provided by the OS
- Usually written in **high-level languages** (C or C++)
- Programs normally access them through **APIs** rather than direct calls
  - Win32 API (Windows)
  - POSIX API (UNIX, Linux, macOS)
  - Java API (JVM)
- **Implementation**
  - Each system call has a unique **number**
  - System-call interface uses a table indexed by these numbers
  - OS kernel executes the requested service and returns results
- **Abstraction**
  - Programmer uses API without knowing internal implementation
  - Runtime libraries handle low-level details



source file

destination file

Example System-Call Sequence

```
Acquire input file name
  Write prompt to screen
  Accept input
Acquire output file name
  Write prompt to screen
  Accept input
Open the input file
  if file doesn't exist, abort
Create output file
  if file exists, abort
Loop
  Read from input file
  Write to output file
Until read fails
Close output file
Write completion message to screen
Terminate normally
```

Figure: A view of operating system services.

Types of System Calls (1)

- **Process Control**
  - create process, terminate process
  - end, abort
  - load, execute
  - get process attributes, set process attributes
  - wait for time
  - wait event, signal event
  - allocate and free memory
  - dump memory if error
  - debugger support, single-step execution
  - locks for managing shared data between processes

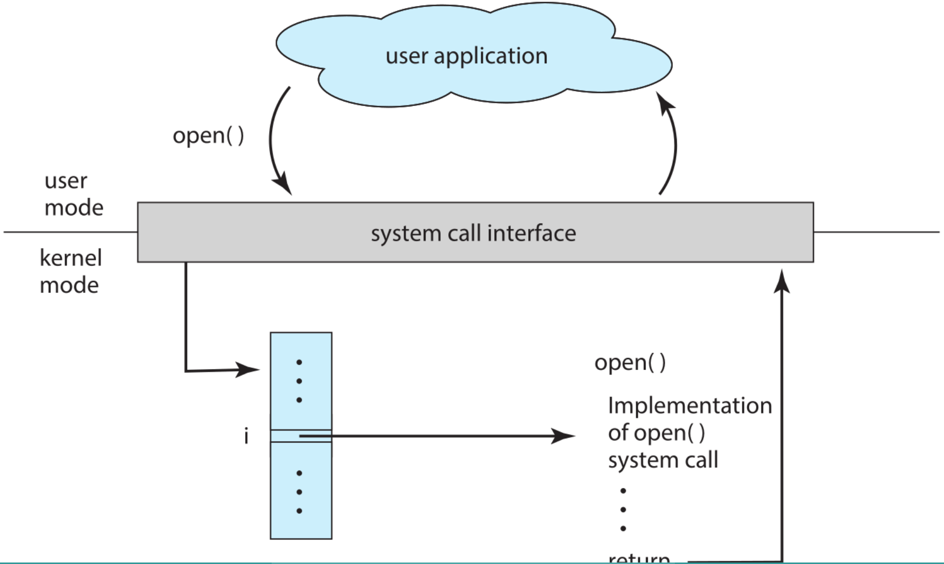
Types of System Calls (2)

- **File Management**
  - create file, delete file
  - open, close file
  - read, write, reposition
  - get and set file attributes
- **Device Management**
  - request device, release device
  - read, write, reposition
  - get device attributes, set device attributes
  - logically attach or detach devices

Types of System Calls (3)

- **Information Maintenance**
  - get time or date, set time or date
  - get system data, set system data
  - get and set process, file, or device attributes
- **Communications**
  - create, delete communication connection
  - send and receive messages (message passing)
  - communication between client and server
  - shared memory creation and access
  - transfer status information
  - attach and detach remote devices
- **Protection**
  - control access to resources
  - get and set permissions
  - allow or deny user access

API – System Call – OS Relationship



System Services

System Services (Overview)

- **System programs** provide a convenient environment for program development and execution
- They are typically grouped into:
  - **File manipulation**
  - **Status information**
  - **Programming language support**
  - **Program loading and execution**
  - **Communications**
  - **Background services**
  - **Application programs**
- For most users, the **view of the operating system** comes from system programs rather than direct system calls
- Some system programs act as simple **interfaces to system calls**, while others are more complex utilities

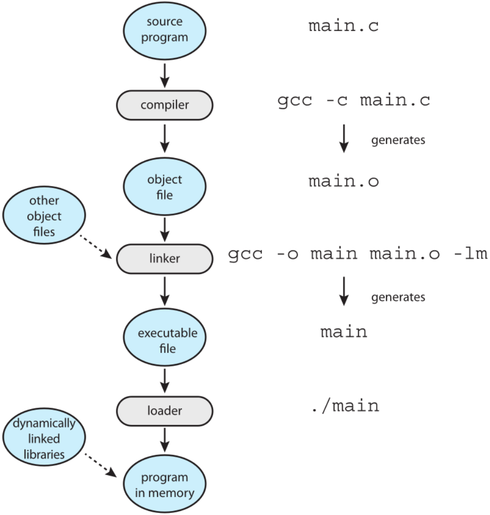
System Services (Examples)

- **File management**
  - Create, delete, copy, rename, print, list, and manipulate files and directories
- **Status information**
  - Provide system information such as date, time, memory, disk space, and users
  - Offer performance monitoring, logging, and debugging information
  - Some systems maintain a **registry** for configuration data
- **File modification**
  - Text editors for creating and editing files
  - Tools for searching or transforming file contents

System Services (Development and Communication)

- **Programming language support**
  - Compilers, assemblers, interpreters, and debuggers
- **Program loading and execution**
  - Absolute loaders, relocatable loaders, linkage editors
  - Overlay loaders and debugging systems
- **Communications**
  - Mechanisms for communication between processes and systems
  - Messaging between users, remote login, file transfer
  - Web browsing and electronic mail

The Role of the Linker and Loader



```
main.c
gcc -c main.c
main.o
gcc -o main main.o -lm
main
./main
```

Figure: The role of the linker and loader.

Why Applications are Operating System Specific?

- Applications compiled for one OS usually **cannot run on another OS**
- Each operating system provides its own
  - **System calls**
  - **File formats** and system interfaces
- Applications can support **multiple operating systems**
  - Written in interpreted languages (Python, Ruby) with interpreters on different OSes
  - Written for a **Virtual Machine** (for example Java JVM)
  - Written in standard languages (like C) and compiled separately for each OS
- **Application Binary Interface (ABI)**
  - Binary-level equivalent of an API
  - Defines how compiled programs interact with the OS and hardware
  - Depends on architecture, CPU, and operating system

References

Avi Silberschatz, Peter Baer Galvin, and Greg Gagne.
Operating system concepts, tenth edition: Lecture slides.
`https://os-book.com/OS10/slide-dir/index.html`, 2018.
Accessed: 2026-03-06.

Galvin P. B. Gagne G. Silberschatz, A.
Operating System Concepts.
Wiley, global edition, 2023.

Dr. Laltu Sardar, Assistant Professor, IISER Thiruvananthapuram
`laltu.sardar@iisertvm.ac.in`, `laltu.sardar.crypto@gmail.com`
Course webpage: https://laltu-sardar.github.io/courses/corgos_2026.html.