

Operating System Overview

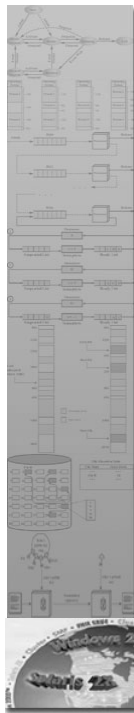
Chapter 2

Muhammad Adri, MT

1

Operating System

- A program that controls the execution of application programs
- An interface between applications and hardware



Muhammad Adri, MT

2

Operating System Objectives

- Convenience
 - Makes the computer more convenient to use
- Efficiency
 - Allows computer system resources to be used in an efficient manner
- Ability to evolve
 - Permit effective development, testing, and introduction of new system functions without interfering with service

Muhammad Adri, MT

3

Layers of Computer System

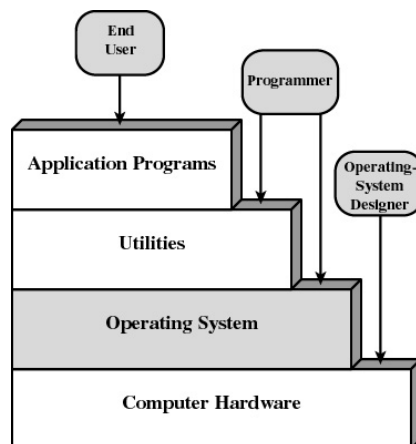
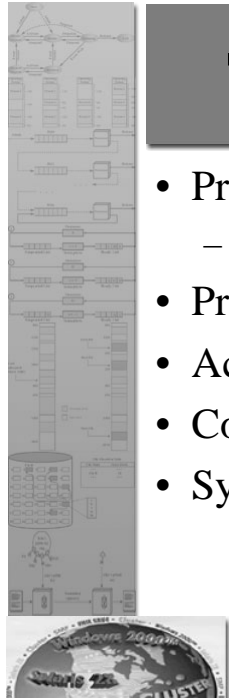


Figure 2.1 Layers and Views of a Computer System

4

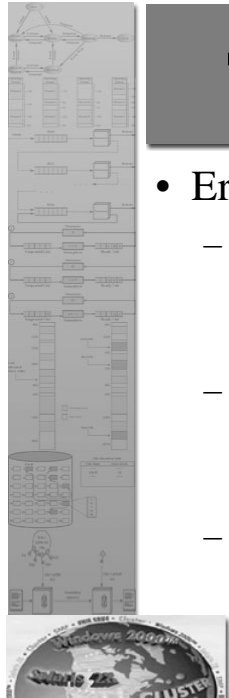


Services Provided by the Operating System

- Program development
 - Editors and debuggers
- Program execution
- Access to I/O devices
- Controlled access to files
- System access

Muhammad Adri, MT

5

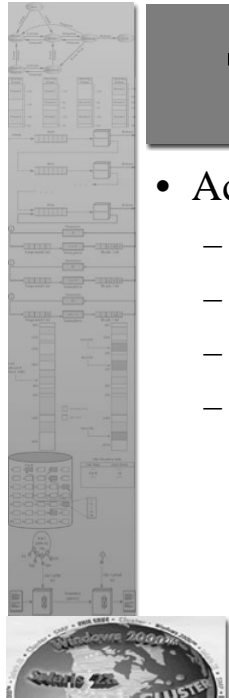


Services Provided by the Operating System

- Error detection and response
 - internal and external hardware errors
 - memory error
 - device failure
 - software errors
 - arithmetic overflow
 - access forbidden memory locations
 - operating system cannot grant request of application

Muhammad Adri, MT

6

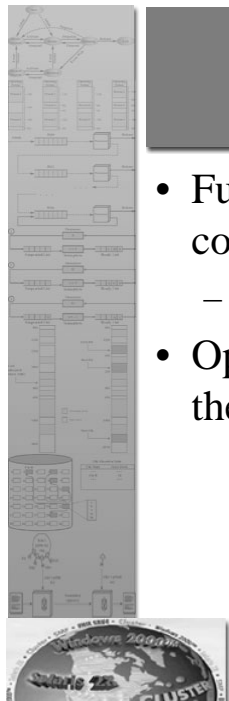


Services Provided by the Operating System

- Accounting
 - collect statistics
 - monitor performance
 - used to anticipate future enhancements
 - used for billing users

Muhammad Adri, MT

7



Operating System

- Functions same way as ordinary computer software
 - It is program that is executed
- Operating system relinquishes control of the processor to execute other programs

Muhammad Adri, MT

8

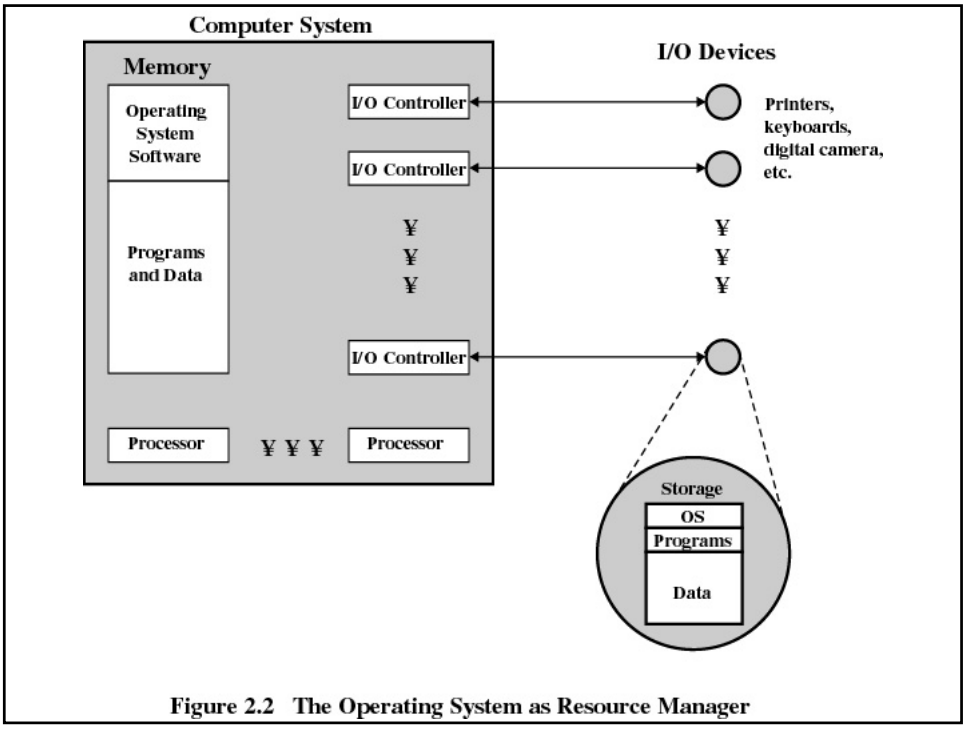


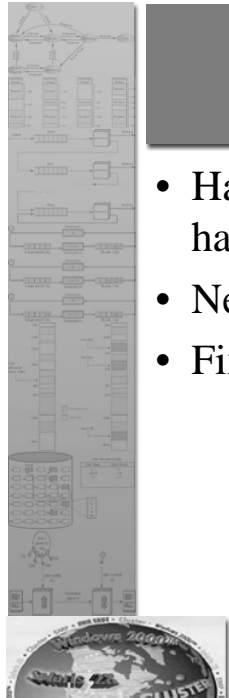
Figure 2.2 The Operating System as Resource Manager

Kernel

- Portion of operating system that is in main memory
- Contains most-frequently used functions
- Also called the nucleus

Muhammad Adri, MT

10

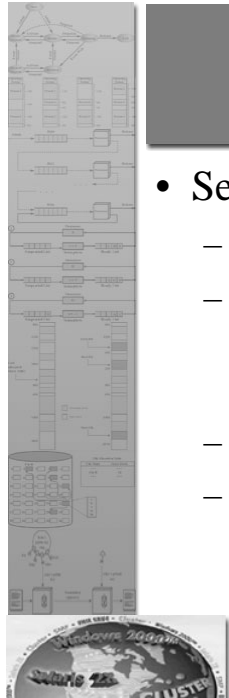


Evolution of an Operating System

- Hardware upgrades and new types of hardware
- New services
- Fixes

Muhammad Adri, MT

11

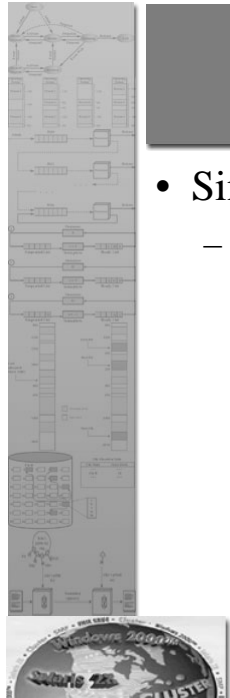


Evolution of Operating Systems

- Serial Processing
 - No operating system
 - Machines run from a console with display lights and toggle switches, input device, and printer
 - Schedule tome
 - Setup included loading the compiler, source program, saving compiled program, and loading and linking

Muhammad Adri, MT

12

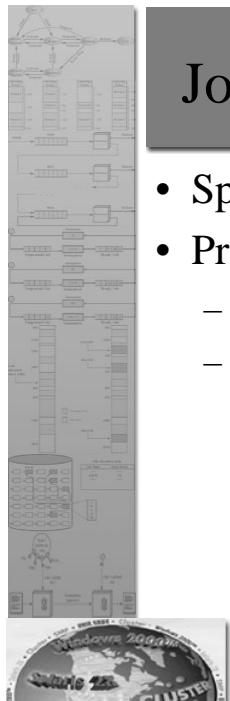


Evolution of Operating Systems

- Simple Batch Systems
 - Monitors
 - Software that controls the running programs
 - Batch jobs together
 - Program branches back to monitor when finished
 - Resident monitor is in main memory and available for execution

Muhammad Adri, MT

13


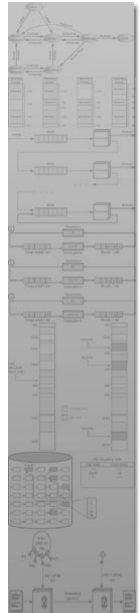


Job Control Language (JCL)

- Special type of programming language
- Provides instruction to the monitor
 - what compiler to use
 - what data to use

Muhammad Adri, MT

14


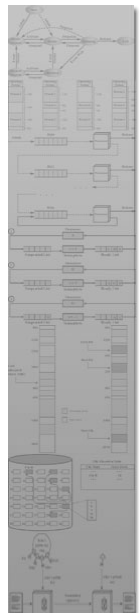


Hardware Features

- Memory protection
 - do not allow the memory area containing the monitor to be altered
- Timer
 - prevents a job from monopolizing the system

Muhammad Adri, MT

15



Hardware Features

- Memory protection
 - do not allow the memory area containing the monitor to be altered
- Timer
 - prevents a job from monopolizing the system

Muhammad Adri, MT

16

Uniprogramming

- Processor must wait for I/O instruction to complete before proceeding

Program A

Time →

(a) Uniprogramming

Muhammad Adri, MT

17

The diagram shows a horizontal timeline for Program A. It starts with a 'Run' box, followed by a 'Wait' period, then another 'Run' box, and finally another 'Wait' period. An arrow labeled 'Time' points to the right below the timeline.

Multiprogramming

- When one job needs to wait for I/O, the processor can switch to the other job

Program A

Program B

Combined

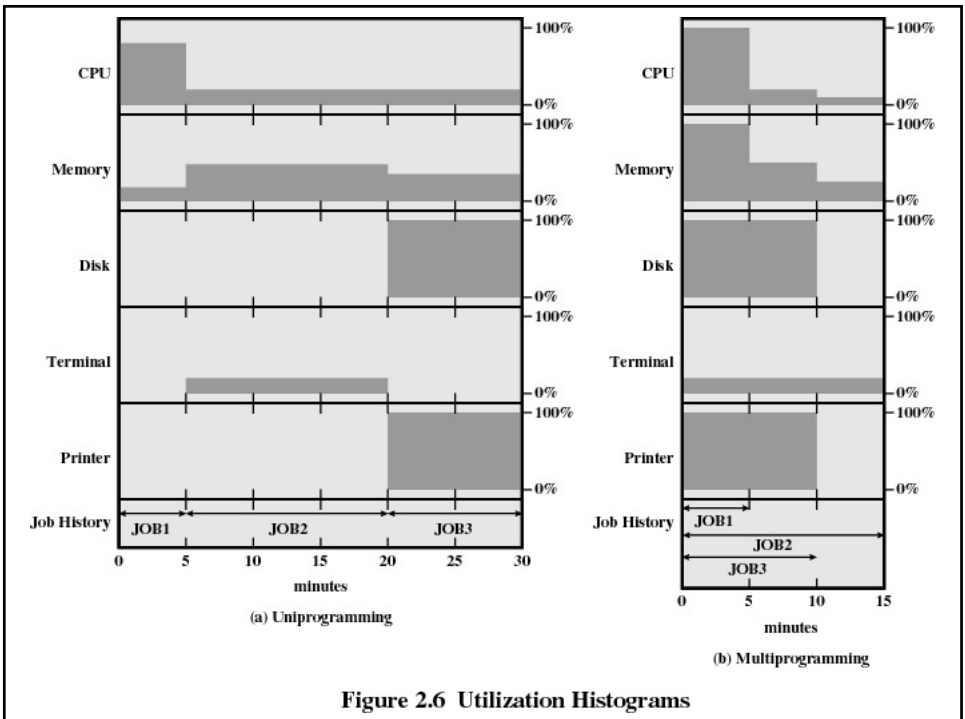
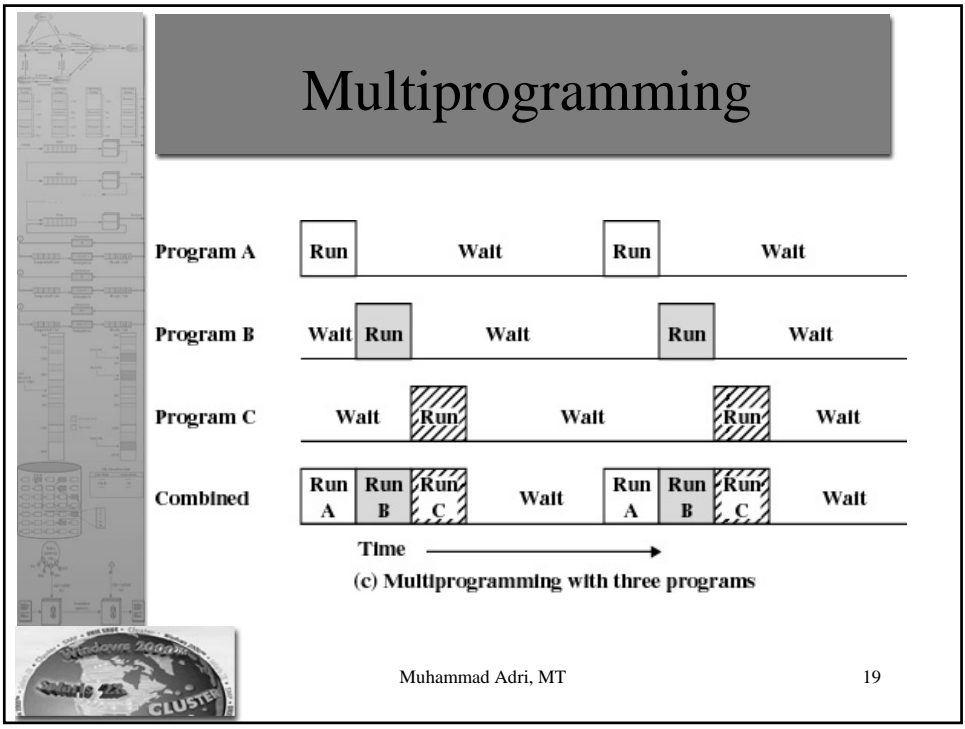
Time →

(b) Multiprogramming with two programs

Muhammad Adri, MT


18

The diagram illustrates multiprogramming with two programs, A and B. It shows three rows of execution over time. Program A has 'Run' boxes at the beginning and end, with 'Wait' periods in between. Program B has 'Run' boxes during the wait periods of Program A. The 'Combined' row shows the interleaved execution: 'Run A' followed by 'Run B', then a 'Wait' period, then 'Run A' followed by 'Run B', and finally a 'Wait' period. An arrow labeled 'Time' points to the right below the timeline.




Example

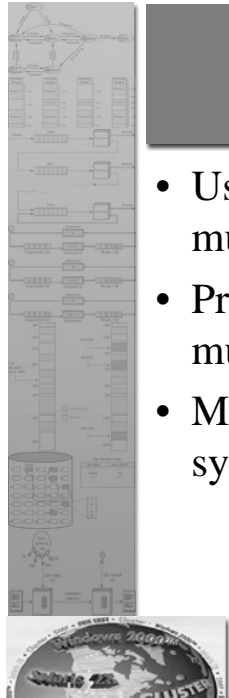
| | JOB1 | JOB2 | JOB3 |
|-----------------|---------------|-----------|-----------|
| Type of job | Heavy compute | Heavy I/O | Heavy I/O |
| Duration | 5 min. | 15 min. | 10 min. |
| Memory required | 50K | 100 K | 80 K |
| Need disk? | No | No | Yes |
| Need terminal | No | Yes | No |
| Need printer? | No | No | Yes |


Muhammad Adri, MT
21

Effects of Multiprogramming

| | Uniprogramming | Multiprogramming |
|--------------------|----------------|------------------|
| Processor use | 22% | 43% |
| Memory use | 30% | 67% |
| Disk use | 33% | 67% |
| Printer use | 33% | 67% |
| Elapsed time | 30 min. | 15 min. |
| Throughput rate | 6 jobs/hr | 12 jobs/hr |
| Mean response time | 18 min. | 10 min. |

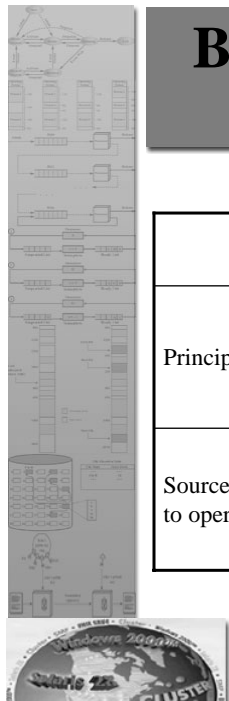

Muhammad Adri, MT
22



Time Sharing

- Using multiprogramming to handle multiple interactive jobs
- Processor's time is shared among multiple users
- Multiple users simultaneously access the system through terminals

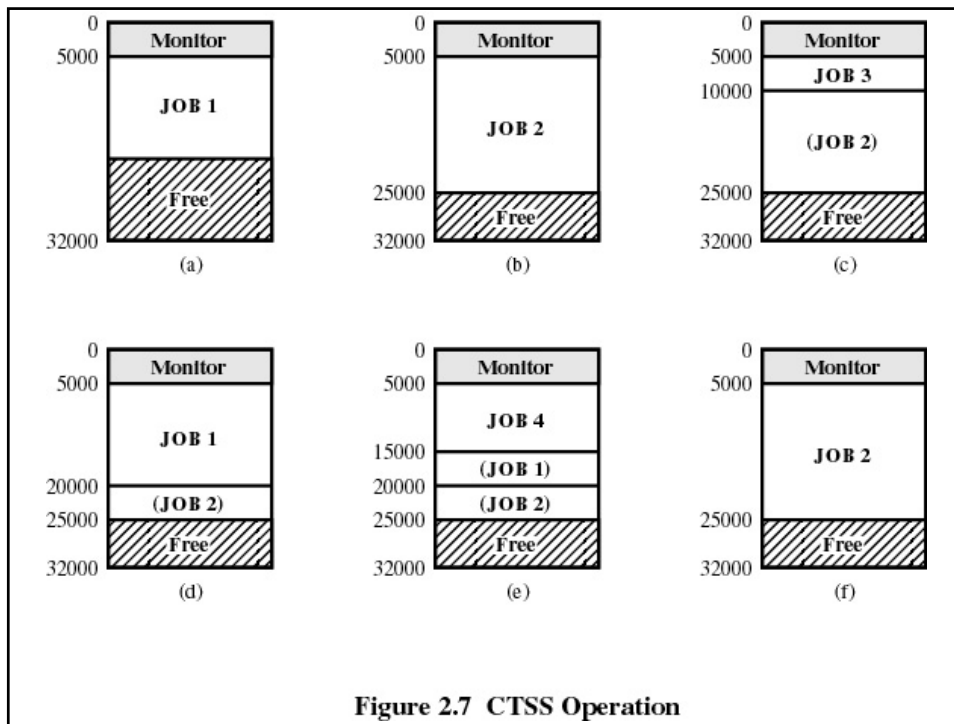
Muhammad Adri, MT 23



Batch Multiprogramming versus Time Sharing

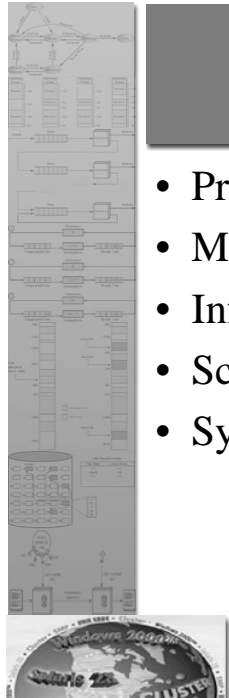
| | Batch Multiprogramming | Time Sharing |
|--|---|----------------------------------|
| Principal objective | Maximize processor use | Minimize response time |
| Source of directives to operating system | Job control language commands provided with the job | Commands entered at the terminal |

Muhammad Adri, MT 24



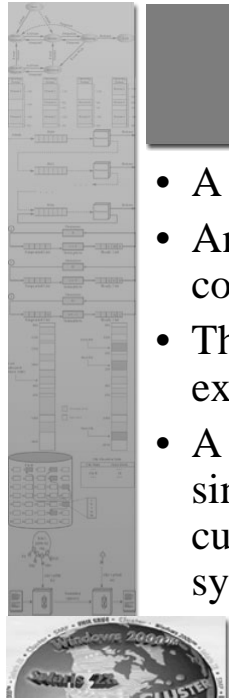
Major Achievements

- Processes
- Memory Management
- Information protection and security
- Scheduling and resource management
- System structure



Muhammad Adri, MT

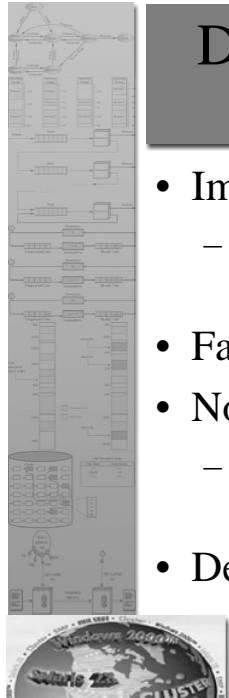
26



Processes

- A program in execution
- An instance of a program running on a computer
- The entity that can be assigned to and executed on a processor
- A unit of activity characterized by a single sequential thread of execution, a current state, and an associated set of system resources

Muhammad Adri, MT 27



Difficulties with Designing System Software

- Improper synchronization
 - ensure a process waiting for an I/O device receives the signal
- Failed mutual exclusion
- Nondeterminate program operation
 - program should only depend on input to it, not relying on common memory areas
- Deadlocks

Muhammad Adri, MT 28

Process

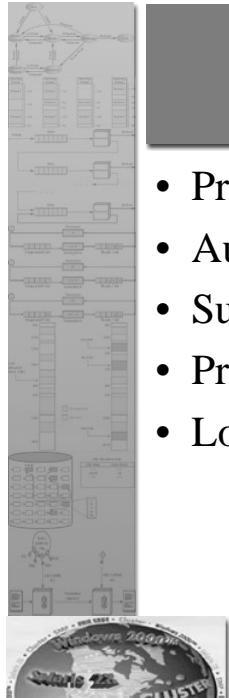
- Consists of three components
 - An executable program
 - Associated data needed by the program
 - Execution context of the program
 - All information the operating system needs to manage the process

Muhammad Adri, MT 29

Process

The diagram illustrates the typical implementation of a process. On the left, a vertical stack represents Main Memory, divided into sections for a Process list (with entries i and j), Process A (containing Context, Data, and Program code), and Process B (containing Context, Data, and Program code). On the right, Processor Registers are shown, including a Process index register pointing to entry i, a PC register, a Base Limit register, and other registers. Arrows indicate the mapping of memory segments to registers: the Program code of Process A is mapped to the PC register, the Data of Process A is mapped to the Base Limit register, and the Context of Process B is mapped to the other registers.

Figure 2.8 Typical Process Implementation 30

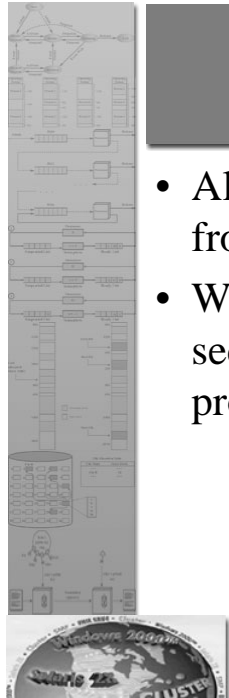


Memory Management

- Process isolation
- Automatic allocation and management
- Support for modular programming
- Protection and access control
- Long-term storage

Muhammad Adri, MT

31

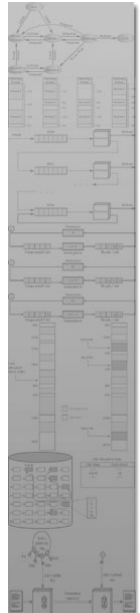


Virtual Memory

- Allows programmers to address memory from a logical point of view
- While one process is written out to secondary store and the successor process read in there in no hiatus


Muhammad Adri, MT

32



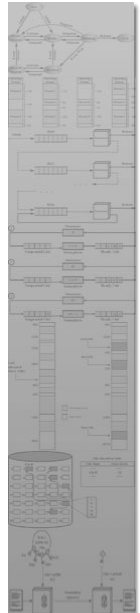
File System

- Implements long-term store
- Information stored in named objects called files




Muhammad Adri, MT

33



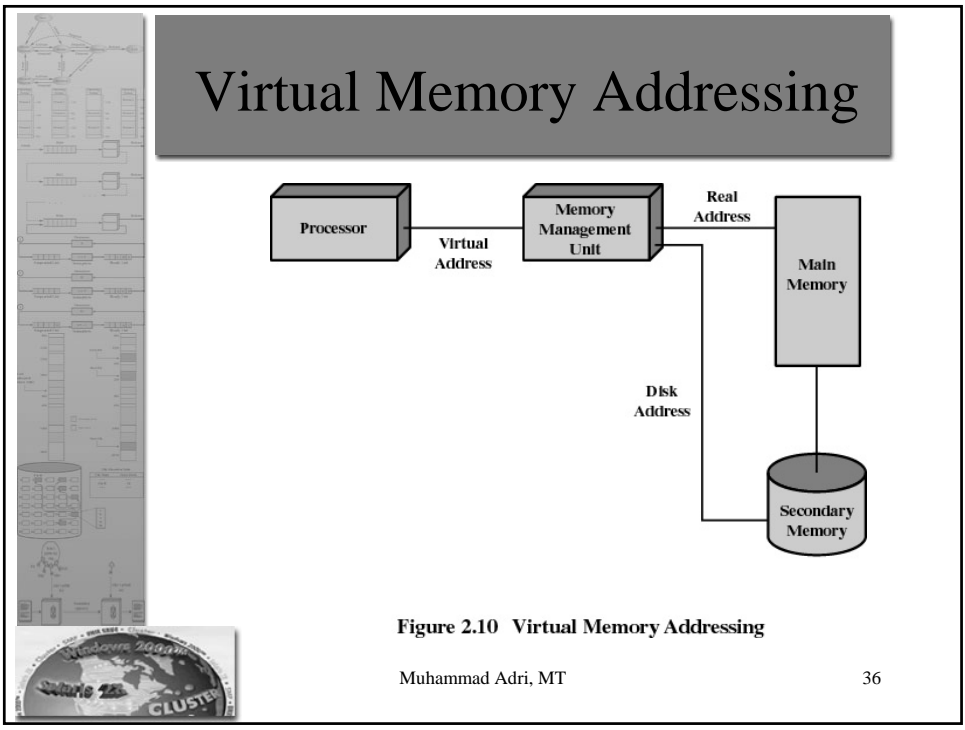
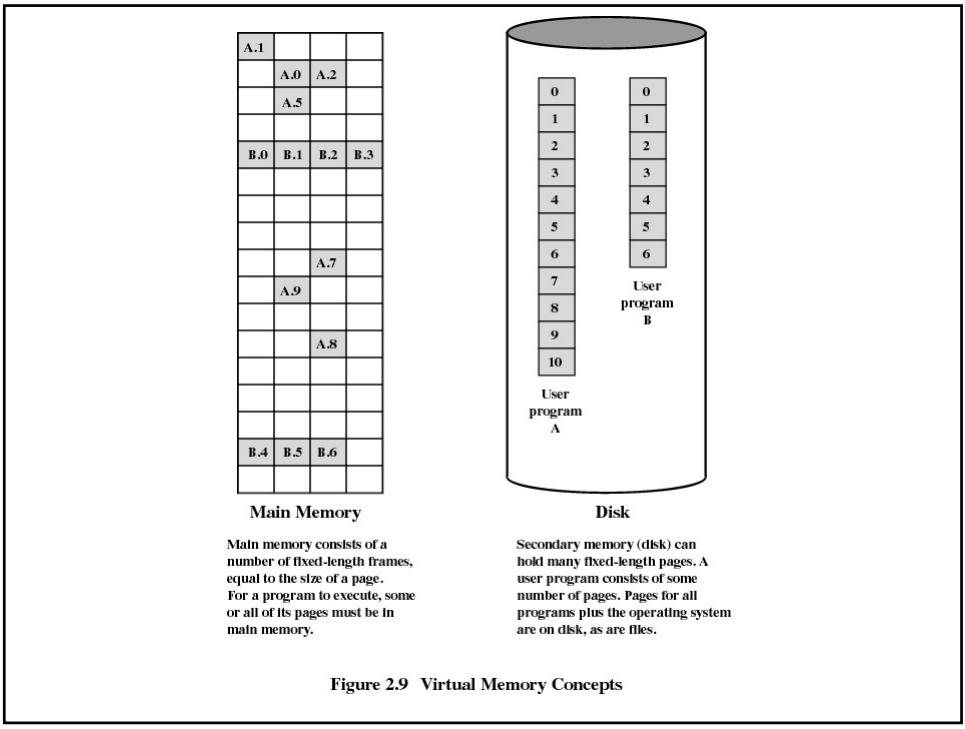
Paging

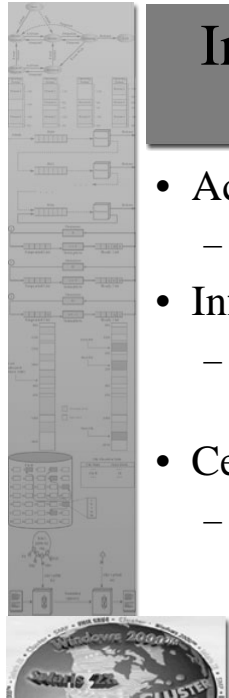
- Allows process to be comprised of a number of fixed-size blocks, called pages
- Virtual address is a page number and an offset within the page
- Each page may be located any where in main memory
- Real address or physical address in main memory



Muhammad Adri, MT

34






Information Protection and Security

- Access control
 - regulate user access to the system
- Information flow control
 - regulate flow of data within the system and its delivery to users
- Certification
 - proving that access and flow control perform according to specifications

Muhammad Adri, MT

37

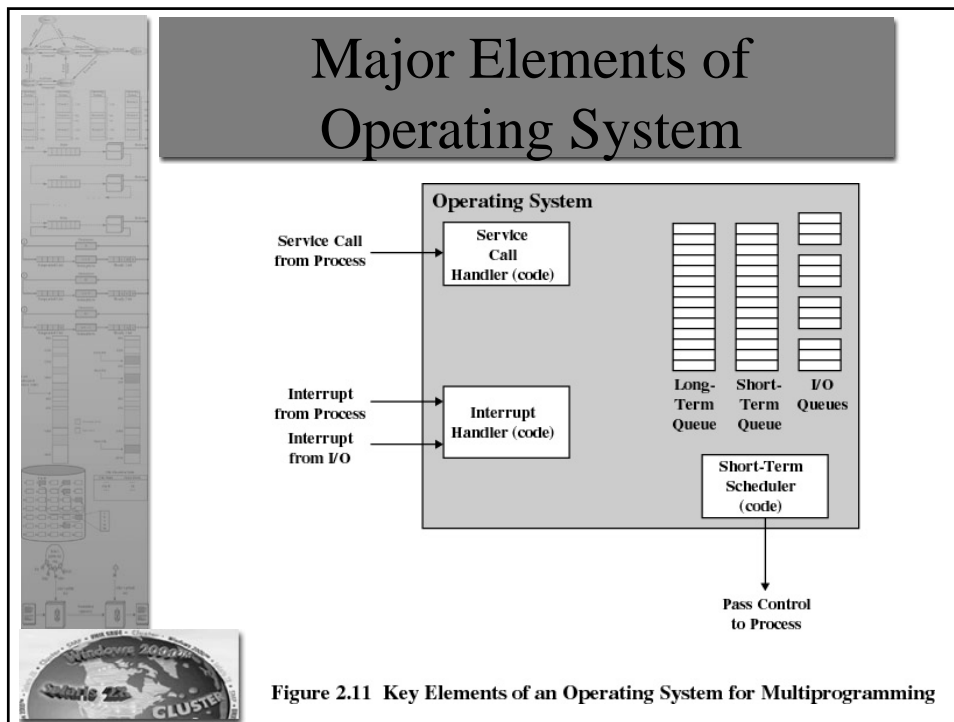


Scheduling and Resource Management

- Fairness
 - give equal and fair access to all processes
- Differential responsiveness
 - discriminate between different classes of jobs
- Efficiency
 - maximize throughput, minimize response time, and accommodate as many users as possible

Muhammad Adri, MT

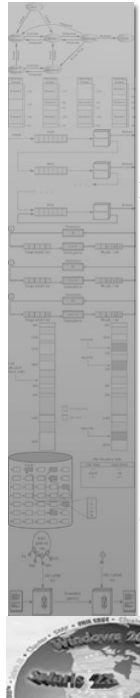
38



System Structure

- View the system as a series of levels
- Each level performs a related subset of functions
- Each level relies on the next lower level to perform more primitive functions
- This decomposes a problem into a number of more manageable subproblems

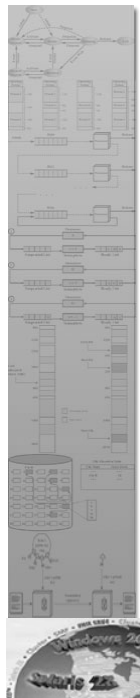
Muhammad Adri, MT 40



Operating System Design Hierarchy

| Level | Name | Objects | Example Operations |
|-------|----------------|---|---|
| 13 | Shell | User programming environment | Statements in shell language |
| 12 | User processes | User processes | Quit, kill, suspend, resume |
| 11 | Directories | Directories | Create, destroy, attach, detach, search, list |
| 10 | Devices | External devices, such as printer, displays and keyboards | Open, close, read, write |
| 9 | File system | Files | Create, destroy, open, close read, write |
| 8 | Communications | Pipes | Create, destroy, open, close, read, write |

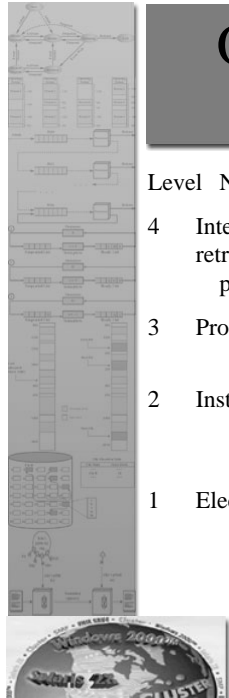
Muhammad Adri, MT 41



Operating System Design Hierarchy

| Level | Name | Objects | Example Operations |
|-------|-----------------------|---|-------------------------------|
| 7 | Virtual Memory | Segments, pages | Read, write, fetch |
| 6 | Local secondary store | Blocks of data, device channels | Read, write, allocate, free |
| 5 | Primitive processes | Primitive process, semaphores, ready list | Suspend, resume, wait, signal |

Muhammad Adri, MT 42

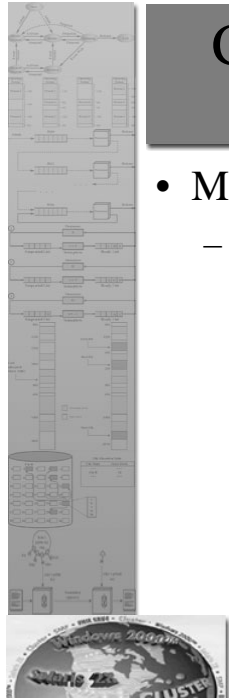


Operating System Design Hierarchy

| Level | Name | Objects | Example Operations |
|-------|---------------------------------|---|--|
| 4 | Interrupts retry programs | Interrupt-handling | Invoke, mask, unmask, |
| 3 | Procedures | Procedures, call stack, display | Mark stack, call, return |
| 2 | Instruction Set | Evaluation stack, micro- program interpreter, scalar and array data | Load, store, add, subtract branch |
| 1 | Electronic circuits | Registers, gates, buses, etc. | Clear, transfer, activate, complement |

Muhammad Adri, MT

43




Characteristics of Modern Operating Systems

- Microkernel architecture
 - assigns only a few essential functions to the kernel
 - address space
 - interprocess communication (IPC)
 - basic scheduling

Muhammad Adri, MT

44

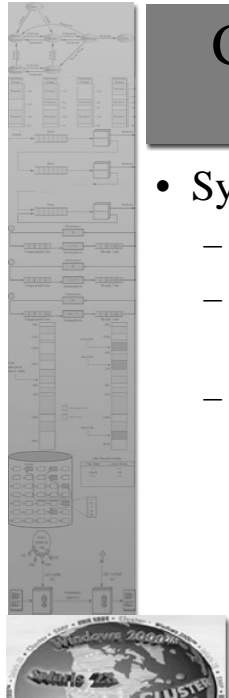


Characteristics of Modern Operating Systems

- **Multithreading**
 - process is divided into threads that can run simultaneously
- **Thread**
 - dispatchable unit of work
 - executes sequentially and is interruptable
- **Process is a collection of one or more threads**

Muhammad Adri, MT

45

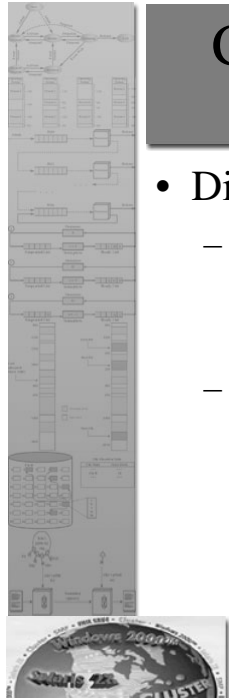


Characteristics of Modern Operating Systems

- **Symmetric multiprocessing**
 - there are multiple processors
 - these processors share same main memory and I/O facilities
 - All processors can perform the same functions

Muhammad Adri, MT

46

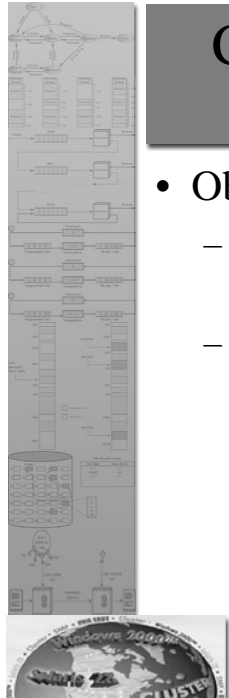


Characteristics of Modern Operating Systems

- Distributed operating systems
 - provides the illusion of a single main memory and single secondary memory space
 - used for distributed file system

Muhammad Adri, MT

47

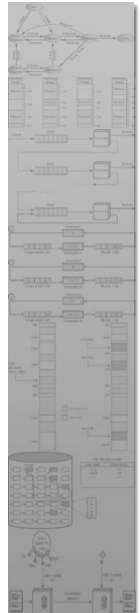


Characteristics of Modern Operating Systems

- Object-oriented design
 - used for adding modular extensions to a small kernel
 - enables programmers to customize an operating system without disrupting system integrity


Muhammad Adri, MT

48



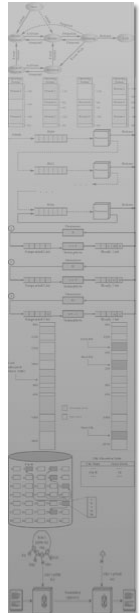
Windows 2000

- Exploits the power of today's 32-bit microprocessors
- Provides full multitasking in a single-user environment
- Client/Server computing




Muhammad Adri, MT

49




Windows 2000 Architecture

- Modular structure for flexibility
- Executes on a variety of hardware platforms
- Supports application written for a variety of other operating system



Muhammad Adri, MT

50

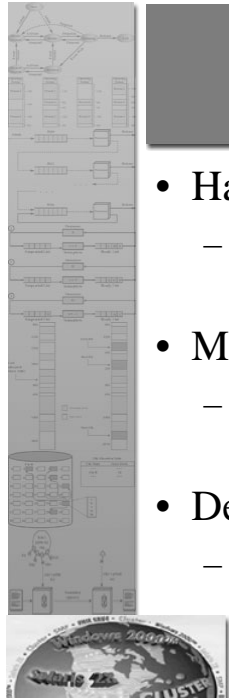


OS Organization

- **Modified microkernel architecture**
 - Not a pure microkernel
 - Many system functions outside of the microkernel run in kernel mode
- **Any module can be removed, upgraded, or replaced without rewriting the entire system**

Muhammad Adri, MT

51

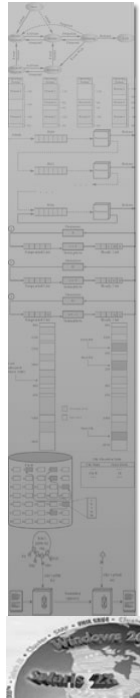


Layered Structure

- **Hardware abstraction layer (HAL)**
 - Isolates the operating system from platform-specific hardware differences
- **Microkernel**
 - Most-used and most fundamental components of the operating system
- **Device drivers**
 - Translate user I/O function calls into specific hardware device I/O requests

Muhammad Adri, MT

52

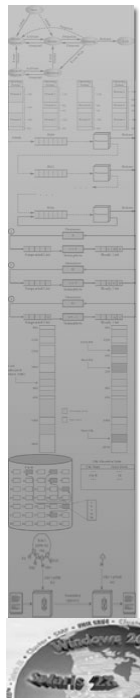


W2K Executive

- I/O manager
- Object manager
- Security reference monitor
- Process/thread manager
- Local procedure call (LPC) Facility
- Virtual memory manager
- Cache manager
- Windows/graphics modules

Muhammad Adri, MT

53

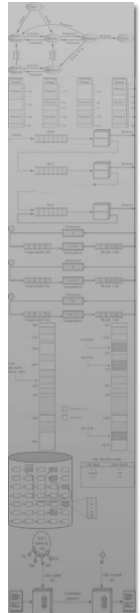


User Processes

- Special system support processes
 - Ex: logon process and the session manager
- Server processes
- Environment subsystems
- User applications


Muhammad Adri, MT

54

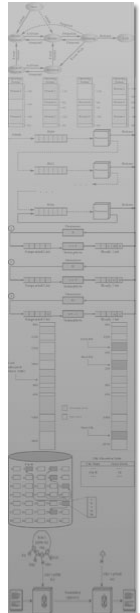


Client/Server Model

- Simplifies the Executive
 - possible to construct a variety of APIs
- Improves reliability
 - each service runs as a separate process with its own partition of memory
 - clients cannot not directly access hardware
- Provides a uniform means fro applications to communicate via LPC
- Provides base for distributed computing




Muhammad Adri, MT 55

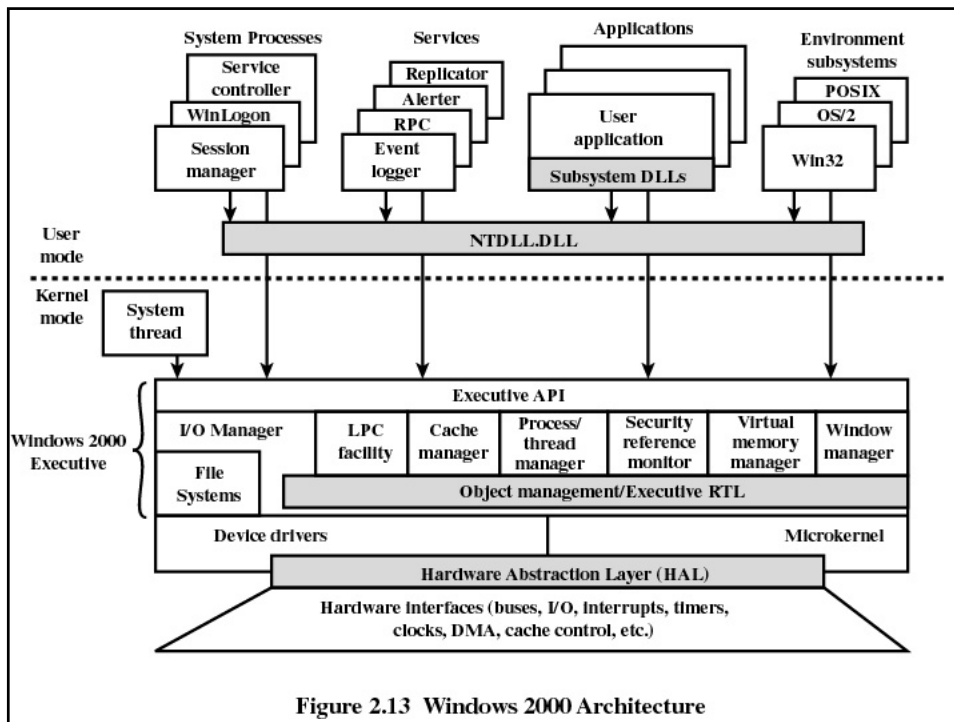


Threads and SMP

- Different routines can execute simultaneously on different processors
- Multiple threads of execution within a single process may execute on different processors simultaneously
- Server processes may use multiple threads
- Share data and resources between process



Muhammad Adri, MT 56



UNIX

- Hardware is surrounded by the operating-system
- Operating system is called the kernel
- Comes with a number of user services and interfaces
 - shell
 - C compiler

Muhammad Adri, MT

58

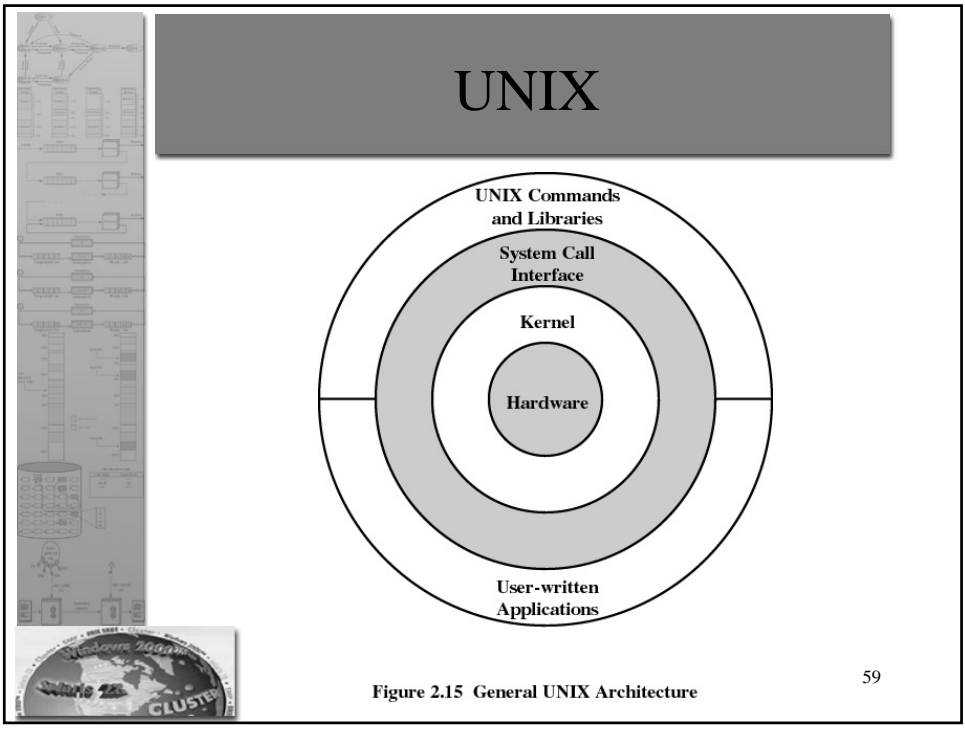


Figure 2.15 General UNIX Architecture

The slide is titled 'Modern UNIX Systems' in a large box at the top. Below the title is a bulleted list of operating systems: System V Release 4 (SVR4), Solaris 2.x, 4.4BSD, and Linux. On the left side of the slide, there is a vertical strip containing a network diagram, a server rack, and a CD-ROM labeled 'Solaris 2.x CLUSTER'.