

Vocational English
III (Mesleki Yabancı
Dil III)
Week 5

22.10.2024



Engineering Faculty
Computer Engineering

Prepared by: Dr Ercan Ezin

This week we will work on

**ENGLISH FOR OPERATING SYSTEMS
&
READING-LISTENING ACTIVITIES**

INTRODUCTION TO OPERATING SYSTEMS

VOCABULARY-I

Kernel (noun) →



Example: "The kernel is responsible for managing system resources and communication between hardware and software."

Process (noun) →



Example: "Each program running on your computer is considered a separate process by the operating system."

Thread (noun) →



Example: "A process can have multiple threads, each performing different tasks."

INTRODUCTION TO OPERATING SYSTEMS

VOCABULARY-2

Scheduler (noun) → 

Example: "The scheduler decides which process runs at any given time."

Memory (noun) → 

Example: "Operating systems manage memory to allocate space for running programs."

File System (noun) → 

Example: "The file system organizes and stores data on a disk."

INTRODUCTION TO OPERATING SYSTEMS

VOCABULARY-3

Driver (noun) →

Example: "Device drivers allow the operating system to communicate with hardware components."

Multitasking (noun) →

Example: "Modern operating systems support multitasking, allowing several programs to run simultaneously."

Boot (verb) →

Example: "The operating system boots the computer by loading essential files from the hard drive."

INTRODUCTION TO OPERATING SYSTEMS

VOCABULARY-4

Interface (noun) → 

Example: "The user interface provides a way for users to interact with the operating system."

Virtual Memory (noun) → 

Example: "Virtual memory allows the operating system to use disk space as additional RAM."

Interrupt (noun) → 

Example: "An interrupt is a signal that informs the operating system that an event needs immediate attention."

INTRODUCTION TO OPERATING SYSTEMS

VOCABULARY-5

Buffer (noun) →



Example: "The buffer temporarily holds data before it is processed or transferred."

Paging (noun) →



Example: "Paging is a memory management scheme used to handle large amounts of data."

Deadlock (noun) →




Example: "A deadlock occurs when two or more processes are unable to proceed because each is waiting for the other to release resources."

INTRODUCTION TO OPERATING SYSTEMS


VOCABULARY-6

Semaphore (noun) → 

Example: "Semaphores are used to control access to shared resources in concurrent processing."

Cache (noun) → 

Example: "The CPU uses a cache to store frequently accessed data for faster processing."

Queue (noun) → 

Example: "The operating system uses a queue to manage tasks waiting for CPU time."

INTRODUCTION TO OPERATING SYSTEMS

VOCABULARY-7

Permissions (noun) → 

Example: "File permissions determine who can read, write, or execute a file."

Command (noun) → 

Example: "Users interact with the system through a command-line interface by typing commands."

Shell (noun) → 


Example: "The shell interprets user commands and communicates them to the operating system."

INTRODUCTION TO OPERATING SYSTEMS

VOCABULARY-8

Fork (verb) → 

Example: "When a process forks, it creates a new process that is a copy of itself."

Daemon (noun) → 


Example: "A daemon is a background process that performs tasks without user interaction."

Swap (verb) → 

Example: "The operating system swaps data between RAM and the hard drive to manage memory."

INTRODUCTION TO OPERATING SYSTEMS

VOCABULARY-9

Privilege (noun) → 

Example: "Operating systems assign different privilege levels to users and processes for security."

Load (verb) → 

Example: "When you open an application, the operating system loads it into memory."

Core (noun) → 

Example: "Modern CPUs have multiple cores, allowing them to run several processes simultaneously."

INTRODUCTION TO OPERATING SYSTEMS

VOCABULARY-6

Latency (noun) →

Example: "Low latency is critical for real-time operating systems used in embedded systems."

Threading (noun) →

Example: "Threading allows a program to run multiple tasks concurrently within the same process."

Reboot (verb) →

Example: "If your system becomes unresponsive, you may need to reboot the computer."

READING COMPREHENSION IN OPERATING SYSTEMS



READING COMPREHENSION IN OS-1

Kernel Documentation

The screenshot shows a web browser window with the URL `kernel.org/doc/html/v4.10/index.html`. The page title is "The Linux Kernel 4.10.0". The left sidebar contains a search bar and a list of navigation links: "The Linux kernel user's and administrator's guide", "Working with the kernel development community", "Development tools for the kernel", "How to write kernel documentation", "The Linux driver implementer's API guide", "Core API Documentation", "Linux Media Subsystem Documentation", "Linux GPU Driver Developer's Guide", "Security documentation", "Linux Sound Subsystem Documentation", "Linux Kernel Crypto API", and "Korean translations". The main content area has a breadcrumb "Docs » Welcome to The Linux Kernel's documentation" and a heading "Welcome to The Linux Kernel's documentation". Below the heading is a paragraph: "This is the top level of the kernel's documentation tree. Kernel documentation, like the kernel itself, is very much a work in progress. We have tried to put all the documents into a coherent whole. Please note that improvements to the documentation are welcome; join the linux-doc list". A section titled "User-oriented documentation" follows, with a paragraph: "The following manuals are written for *users* of the kernel — those who are trying to get it to work optimally on a given system". A bulleted list of links is provided: "The Linux kernel user's and administrator's guide" (with sub-links for "Linux kernel release 4.x", "The kernel's command-line parameters", "Linux allocated devices (4.x+ version)", "Reporting bugs", "Security bugs", "Bug hunting", "Bisecting a bug", "Tainted kernels", "Ramoops oops/panic logger", "Dynamic debug", and "Explaining the dreaded 'No init found.' boot hang message").

kernel.org/doc/html/v4.10/index.html

🏠 The Linux Kernel
4.10.0

Search docs

The Linux kernel user's and administrator's guide

Working with the kernel development community

Development tools for the kernel

How to write kernel documentation

The Linux driver implementer's API guide

Core API Documentation

Linux Media Subsystem Documentation

Linux GPU Driver Developer's Guide

Security documentation

Linux Sound Subsystem Documentation

Linux Kernel Crypto API

Korean translations

Docs » Welcome to The Linux Kernel's documentation

Welcome to The Linux Kernel's documentation

This is the top level of the kernel's documentation tree. Kernel documentation, like the kernel itself, is very much a work in progress. We have tried to put all the documents into a coherent whole. Please note that improvements to the documentation are welcome; join the linux-doc list

User-oriented documentation

The following manuals are written for *users* of the kernel — those who are trying to get it to work optimally on a given system

- [The Linux kernel user's and administrator's guide](#)
 - [Linux kernel release 4.x](#) <<http://kernel.org/>>
 - [The kernel's command-line parameters](#)
 - [Linux allocated devices \(4.x+ version\)](#)
 - [Reporting bugs](#)
 - [Security bugs](#)
 - [Bug hunting](#)
 - [Bisecting a bug](#)
 - [Tainted kernels](#)
 - [Ramoops oops/panic logger](#)
 - [Dynamic debug](#)
 - [Explaining the dreaded "No init found." boot hang message](#)

READING COMPREHENSION IN OS-1

Windows Architecture Documentation

The screenshot shows a web browser window with the URL `learn.microsoft.com/en-us/sysinternals/resources/windows-internals`. The page title is "Windows Internals Book". The left sidebar contains a navigation menu with categories like Home, Downloads, Sysinternals Suite, Microsoft Store, Community, and Resources. The "Resources" section is expanded, showing "Windows Internals Book" as the selected item. The main content area features the title "Windows Internals Book", a sub-header "Article • 09/15/2022 • 3 contributors", and a "Feedback" button. Below this is a section titled "In this article" with a list of links: "Table of contents of the 7th edition, part 1:", "History of the Book", "Seventh Edition Changes", and "Book tools". The main text begins with "Windows Internals 7th edition (Part 1) covers the architecture and core internals of Windows 10 and Windows Server 2016. This book helps you:" followed by a bulleted list of six key topics.

learn.microsoft.com/en-us/sysinternals/resources/windows-internals

Sysinternals Downloads Community Resources

Filter by title

Learn / Sysinternals /

Windows Internals Book

Article • 09/15/2022 • 3 contributors [Feedback](#)

In this article

- [Table of contents of the 7th edition, part 1:](#)
- [History of the Book](#)
- [Seventh Edition Changes](#)
- [Book tools](#)

Windows Internals 7th edition (Part 1) covers the architecture and core internals of Windows 10 and Windows Server 2016. This book helps you:

- Understand the Windows system architecture and its general components
- Explore internal data structures using tools like the kernel debugger
- Understand how Windows uses processes for management and isolation
- Understand and view thread scheduling and how CPU resources are managed
- Dig into the Windows security model including recent advances in security mitigations
- Understand how Windows manages virtual and physical memory
- Understand how the I/O system manages physical devices and device drivers

READING COMPREHENSION IN OS-1

Virtualisation Documentation

The screenshot shows the VMware Docs website for VMware vSphere. The page title is "VMware vSphere Documentation". The left sidebar contains a navigation menu with the following items:

- Expand All
- vSphere 8.0
 - ESXi and vCenter
 - vSphere IaaS Control Plane
 - VMware vSAN
 - SDK and API Documentation
 - CLI Documentation
- vSphere 7.0
 - ESXi and vCenter Server
 - vSphere with Tanzu
 - VMware vSAN
 - vSphere Bitfusion
 - SDK and API Documentation
 - CLI Documentation
- vSphere 6.7
 - ESXi and vCenter Server
 - vSphere Update Manager
 - VMware vSAN

The main content area features a "Feedback" link and a paragraph describing VMware vSphere: "VMware vSphere is VMware's virtualization platform, which transforms data centers into aggregated computing infrastructures that include CPU, storage, and networking resources. vSphere manages these infrastructures as a unified operating environment, and provides you with the tools to administer the data centers that participate in that environment."

Below the text is a diagram illustrating the VMware vSphere architecture. The diagram shows a central "vCenter Server" box connected to a "vSphere Client" box above it. A line labeled "Enhanced Linked Mode" connects the vSphere Client to a vertical stack of three "vCenter Server" boxes on the right. The central vCenter Server is connected to a "Manage" line that points to a dashed box containing two "ESXi" boxes. Each ESXi box has three "vm" icons above it. Below the ESXi boxes are two server icons connected to a central storage icon. The entire diagram is enclosed in a dashed blue border.

By clicking accept, you understand that we use cookies to improve

READING COMPREHENSION IN OS-1

Distributed OS Documentation

The screenshot shows a web browser at the URL `javatpoint.com/distributed-operating-system`. The page features a navigation menu with links for Home, Python, Java, JavaScript, HTML, SQL, PHP, C#, C++, DS, Aptitude, Reasoning, and Selenium. The main content area is titled "Distributed Operating System" and includes a "prev" and "next" button. The text describes a distributed operating system (DOS) as an essential type of operating system that uses many central processors to serve multiple real-time applications and users. It connects multiple computers via a single communication channel, and each system has its own processor and memory. These CPUs communicate via high-speed buses or telephone lines. Individual systems that communicate via a single channel are regarded as a single entity and are also known as loosely coupled systems.

The diagram illustrates a distributed system with four nodes, each labeled "CPU & Memory", arranged in a circle around a central "Communication Network".

This operating system consists of numerous computers, nodes, and sites joined together via LAN/WAN lines. It enables the

READING COMPREHENSION IN OS-1

Real Time OS Documentation

The screenshot shows a web browser window with the URL `freertos.org/Documentation/01-FreeRTOS-quick-start/01-Beginners-guide/01-RTOS-fundamentals`. The page features the FreeRTOS logo and a navigation menu with links for 'About FreeRTOS', 'Documentation', 'Security', 'Partners', and 'Community'. A left sidebar contains a 'DOCUMENTATION' section with a tree view: 'Overview', 'FreeRTOS quick start', 'Beginners guide' (expanded), 'Overview', 'RTOS fundamentals' (highlighted), 'FreeRTOS kernel quick start guide', 'Build your first project', 'FreeRTOS libraries and 3rd party tools', 'FreeRTOS plus AWS solutions', and 'Join the FreeRTOS community'. The main content area shows the breadcrumb 'FreeRTOS quick start > Beginners guide', the update date 'Updated Oct 2024', the title 'RTOS Fundamentals', and the subtitle 'An overview of real-time operating systems'. The 'Introduction' section defines an RTOS as a small, deterministic system used in embedded devices like medical equipment and automotive ECUs, noting that it has benefits even without strict real-time requirements. The 'Multitasking' section states that the kernel is the core component of an OS, allowing multiple users to access the processor simultaneously.

freertos.org/Documentation/01-FreeRTOS-quick-start/01-Beginners-guide/01-RTOS-fundamentals

freeRTOS About FreeRTOS Documentation Security Partners Community

DOCUMENTATION

- Overview
- FreeRTOS quick start
 - Beginners guide
 - Overview
 - RTOS fundamentals**
 - FreeRTOS kernel quick start guide
 - Build your first project
 - FreeRTOS libraries and 3rd party tools
 - FreeRTOS plus AWS solutions
 - Join the FreeRTOS community

Kernel

FreeRTOS quick start > Beginners guide

Updated Oct 2024

RTOS Fundamentals

An overview of real-time operating systems

Introduction

A Real-Time Operating System (RTOS) is a type of computer operating system designed to be small and deterministic. RTOSes are commonly used in embedded systems such as medical devices and automotive ECUs that need to react to external events within strict time constraints. Typically this class of embedded system only has one or two requirements demanding that level of deterministic timing, and using an RTOS has benefits even when the embedded system has no hard real-time requirement at all. See the FAQ "[Why use an RTOS?](#)".

An RTOS is typically smaller and lighter weight than a general purpose operating system, making RTOSes suitable for memory, compute and power constrained devices.

Multitasking

The kernel is the core component within an operating system. General purpose operating systems, such as Linux, employ kernels that allow multiple users to access the computer's processor seemingly simultaneously. These multiple users can each execute multiple programs apparently concurrently.

READING COMPREHENSION IN OS-1

Cloud Based OS Documentation

The screenshot shows a web browser window displaying the Google Cloud documentation page for "Operating system details". The browser's address bar shows the URL "cloud.google.com/compute/docs/images/os-details". The page header includes the Google Cloud logo, navigation links for "Documentation", "Technology areas", "Cross-product tools", and "Related sites", and a search bar. Below the header, there are tabs for "Compute Engine", "Guides", "Reference", "Samples", and "Resources". A left sidebar contains a "Filter" button and a list of navigation items under "Virtual machine instances" and "Get started". The main content area features a breadcrumb trail "Compute Engine > Documentation > Guides", the title "Operating system details" with a bookmark icon, and a "Send feedback" button. Below the title is a "Release Notes" link. The main text explains that the page provides general OS details and feature support for OS images available on Compute Engine. It notes that some OS images are customized for Compute Engine and differ from standard vendor images. A link to the "Support and maintenance policy for OS images" is provided. The section "CentOS" is partially visible at the bottom, with a red warning icon and text stating: "Caution: CentOS operating systems have reached their end of development and support. For more information, see [CentOS EOS guidance](#)."

cloud.google.com/compute/docs/images/os-details

Google Cloud Documentation Technology areas Cross-product tools Related sites Search

Compute Engine Guides Reference Samples Resources

Filter

Virtual machine instances
Instance groups
Machine type families
CPU platforms
GPUs
Regions and zones

Get started
Plan and prepare
Work with regions and zones
Review VM deployment options
Networking overview for VMs
Images and operating systems
OS images
About OS images
Operating system details
OS image lifecycle
Support policy
Premium operating systems
Access control
Name resources

Compute Engine > Documentation > Guides

Operating system details

Send feedback

[Release Notes](#)

This page provides general operating system (OS) details and feature support for the [OS images](#) that are available on Compute Engine.

Some OS images are customized specifically to run on Compute Engine and have notable differences from the standard images that come directly from the operating system vendors. These differences are also covered for each OS.

For information about how support and maintenance is provided for these OS images on Compute Engine, based on support package, license type, and image lifecycle stage, see [Support and maintenance policy for OS images](#).

CentOS

Caution: CentOS operating systems have reached their end of development and support. For more information, see [CentOS EOS guidance](#).

LISTENING

English FOR IT

OPERATING SYSTEMS LISTENING ACTIVITY.

Please go to credit page for video web address.(will be updated later)

ASSIGNMENT

Write a paragraph about **only one of the topic** below with your friend. Up to 3 friends or you can work alone. Send me your findings.

- Compare different operating system architectures (e.g., monolithic vs. microkernel) and explain which one is more efficient in modern systems.
- Research file systems like NTFS and ext4. Write about how they differ in performance and security.
- Write about the role of virtualization in modern operating systems and its impact on resource management.
- Analyze the security features of a popular operating system. Focus on things like user permissions and encryption.
- Research how memory management works in a specific operating system (e.g., Linux or Windows). Discuss how it affects performance.
- Compare open-source operating systems (e.g., Linux) with proprietary ones (e.g., Windows) and write about the pros and cons of each.
- Research real-time operating systems (RTOS) and explain how they differ from general-purpose operating systems.
- Write a review of a specific feature in an operating system (e.g., Windows Task Manager). Explain how it helps users.



EOF*

*End of Fun/File