

Titolo insegnamento (IT)	Sistemi Operativi e Reti
Name of the course (EN)	Operating Systems and Computer Networks
Settore Scientifico Disciplinare (SSD) di Riferimento	Informatica - INF/01
Numero moduli / # of modules	2 (Sistemi Operativi/Operating Systems; Reti di Calcolatori/Computer Networks)
Numero ore di didattica assistita / Course duration	96h: 48 ore di didattica frontale + 48 ore di attività di laboratorio/48h lectures + 48h laboratory
Orario ricevimento studenti / Office hours	Lunedì h15:00 - 17:00
Email del docente / Teacher's e-mail	ianni@unical.it
Telefono/Phone	0984-496430

IT(Italiano)

EN(English)

Obiettivi formativi - Risultati di apprendimento attesi- Learning goals	
1. Conoscenza dell'architettura dei sistemi operativi, con particolare riferimento a gestione di memoria, dispositivi hardware, e unità di calcolo; 2. Capacità di riconoscere il ruolo delle componenti architetturali astratte nei sistemi operativi reali Windows e Linux; 3. Capacità di amministrare sistemi standalone; 4. Conoscenza di un linguaggio di scripting (perl); 5. Conoscenza delle principali metodologie di programmazione concorrente; 6. Conoscenza architetturale delle reti di calcolatori e dei protocolli di rete più comuni, con particolare riferimento allo stack TCP/IP e ai protocolli di livello applicazione principali; 7. Capacità di progettazione e messa in sicurezza di reti LAN	1. Acquiring knowledge of fundamental basics on operating systems architecture, with respect to the memory management aspects, hardware devices, processing units; 2. Discerning the role of abstract architectural components in real operating systems like Windows as well as Linux; 3. Acquiring essential operating systems skills including how to administrate stand-alone systems; 4. Understanding of a scripting language (perl); 5. Understanding of basic methodologies for concurrent programming; 6. Understanding of computer networks architectural principles as well as networks protocols, including the TCP/IP stack and the more common application level protocols ; 7. Acquiring fundamental skills for designing and securing LAN networks;

Prerequisiti / Skills required for attending the course

Conoscenza di un linguaggio di programmazione ad oggetti (preferibilmente Java), e dell'architettura di un calcolatore.	Knowledge of an object oriented programming language (preferably, Java) and of Computer Architecture
---	--

Programma del corso - Course Programme

Parte I (Sistemi operativi)

- Definizione di sistema operativo. Componenti base di un sistema operativo. Meccanismi di interrupt e DMA.
- Concetto di processo e di thread. Operazioni di task switching. Il Program Control Block.
- Diagramma a stati dell'attività di un thread. Multitasking con e senza prelazione. Scheduling dei processi nell'architettura Win32 e Linux.
- Sincronizzazione tra processi. Monitor e altre strutture dati di base in Java. Problemi tipici di sincronizzazione: buffer limitato, lettori e scrittori, problema dei cinque filosofi.
- Situazioni di stallo tra processi. Grafo dell'allocazione delle risorse. Situazioni di starvation.
- Gestione della memoria. Paginazione: politiche di swap in e swap out delle pagine. Protezione della memoria. Caricamento in memoria di programmi. Rilocalizzazione. Chiamata a funzioni di sistema.
- Tool per il monitoraggio dell'attività dei processi Windows e Linux (procmon, procexp, top, ps).
- File Systems: strutture dati di supporto alla gestione di file e directory. Mappatura tra dispositivi fisici e entità su livello di astrazione superiore.
- Linux: layout del file system. Attributi e permessi di accesso. Gestione dei processi.
- Comandi shell Linux: gestione dei file e dei permessi di accesso, piping e redirection. Comandi di gestione dei processi. Avvio di processi in background. Variabili d'ambiente, passaggio di parametri su linea di comando.
- Linguaggi di scripting (Perl)

Parte II (Reti di Calcolatori)

- Concetto di gerarchia di protocolli.
- Livello applicazione: Uso dei socket in Java. Gestione di protocolli di livello applicazione in Java. I protocolli HTTP e SMTP. Application multiplexing. Risoluzione dei nomi e DNS.
- Livello di trasporto: Protocolli TCP e UDP. Protocolli di comunicazione stop & wait e a finestra mobile. Controllo di congestione e di flusso. Protocolli di connessione e disconnessione. Handshaking a tre vie.
- Livello di rete: concetto di qualità del servizio: latenza, jitter, banda, affidabilità. Architetture di interconnessione store & forward. Protocollo IP: sistema di numerazione degli host, struttura di un frammento IP. Tabelle di routing. Cenni alle tecniche

Part I (Operating Systems)

- Definition of Operating System. Key components of an Operating System. Interrupt mechanism and DMA.
- The concepts of Process and Thread. Task switching. The Program Control Block.
- State Diagram for the Life Cycle of Threads. Multitasking with/without prelation. Process scheduling in Win32 and Linux architecture.
- Process synchronization. Monitors and other basic Java data structures. Common synchronization problems: the bounded buffer, the readers-writers, the five philosophers.
- Deadlock Characterization for resources contention between processes. Resource allocation graph. Starvation.
- Memory management. Paging: swap in and out policies. Memory protection and process separation. Program loading. Relocation. System calls.
- System monitoring tools under Windows and Linux (procmon, procexp, top, ps).
- File Systems: data structures for managing files and directories. Mapping of physical devices to logical level entities.
- Linux Shell: file management and access privileges, piping and redirection. Process management. Background processes. Environment variables, command line arguments.
- The Perl Language: constructs, native data structures, regular expressions, interfacing to the host operating system.

Part II (Computer Networks)

- Network Protocol Stacks.
- Application Level: usage of Java sockets. Application Protocols management in Java. The HTTP and the SMTP protocols. Application multiplexing. Name resolution and DNS.
- Transport Level: The TCP and the UDP protocols. The Stop & Wait and the sliding window protocols. Flow and congestion control. Connection and disconnection protocols. 3-way Handshaking.
- Network Level: the Quality of Service concept: latency, jitter, reliability. Store & Forward interconnection architectures. IP Protocol: host numbering scheme, structure of an IP fragment. Routing tables. Dynamic Routing.

di routing dinamico. • Livello 'data link': Protocollo ARP. Protocolli di condivisione del mezzo trasmissivo. Protocolli Ethernet e WiFi 802.11. Struttura di un frame Ethernet e sistema di addressing. Dispositivi di ripetizione del segnale: hub, switch e access point. • Il livello fisico. Cenni di teoria dei segnali. Nozioni di latenza e banda di un canale trasmissivo. Teorema di Nyquist. Principali mezzi trasmissivi. Commutazioni di circuito, di frequenza, di tempo, di pacchetto. • Esercitazioni: le esercitazioni di laboratorio sono principalmente svolte con l'ausilio dei software open source Wireshark (analizzatore del traffico di rete), e Netkit (ambiente per la simulazione di reti virtuali), nonché facendo uso dei tool di sistema forniti dai sistemi operativi Windows e Linux. a. Livello applicazione: programmazione di un client SMTP e di un client HTTP. b. Livello di trasporto: comandi di monitoraggio delle connessioni: netstat. Port scanners. Interpretazione dell'evoluzione dei protocolli più comuni con Ethereal/Wireshark (DNS, HTTP, TCP). c. Livello di rete: interpretazione e configurazione di una tabella di routing con Netkit, concetto di mascheramento e di subnet. Comando route. Protocollo ICMP: ping, tracer. d. Firewalling: configurazione di un firewall sotto linux con iptables. e. Livello data link: monitoraggio e configurazione di una rete ethernet, principali comandi di gestione Windows e Linux: ipconfig, ifconfig, arp. f. Livello fisico: visita agli apparati di rete dipartimentali. Principali mezzi trasmissivi.	• Data link Level: ARP Protocol. Protocols for medium access control. Ethernet and WiFi 802.11 protocols. Structure of an Ethernet frame and its addressing system. Devices for signal repetition: hub, switches and access points. • Physical Level: introduction to the basics of signal theory. The concepts of latency and bandwidth of a transmission channel. The Nyquist theorem. Main transmission media. Circuit switching, Frequency multiplexing, Time multiplexing, Packet switching. Laboratory: Lectures with the use of the open source software Wireshark (a network protocol analyzer), Netkit (an environment for setting up and performing networking experiments), as well as system tools provided by Windows and Linux. Application Level: programming a SMTP client and an HTTP client. Transport Level: commands for monitoring connections : netstat. Port scanners. Analyzing the evolution of common protocols with Ethereal/Wireshark (DNS, HTTP, TCP). Network Level: understanding and configuring a routing table using Netkit; the concepts of mask and subnet. The route command. The ICMP protocol: ping, tracer. Firewalling: configuring a firewall under Linux using iptables. Data Link Level: monitoring and configuring an Ethernet network, common commands provided by Windows and Linux: ipconfig, ifconfig, arp. Physical Level: visit to the departmental network devices. Fundamental transmission media: ether, copper, fiber optic, cable types and connectors.
--	--

Testi di riferimento - Reference material	
• H.M.Deitel, P.J.Deitel, D.R.Choffnes, Operating Systems, 3d Edition, Pearson Education. • Randal L. Schwartz, Tom Phoenix, Brian d. Foy, "Learning Perl", O' Reilly. • J. F. Kurose e K.W. Ross, "Computer Networking: A Top-Down Approach", 6th edition, Pearson Education. • Dispense e materiale didattico fornito dal docente sulla pagina Web del corso.	• H.M.Deitel, P.J.Deitel, D.R.Choffnes, Operating Systems, 3d Edition, Pearson Education. • Randal L. Schwartz, Tom Phoenix, Brian d. Foy, "Learning Perl", O' Reilly. • J. F. Kurose e K.W. Ross, "Computer Networking: A Top-Down Approach", 6th edition, Pearson Education. • Student's material and slides available on the course's web page.

Attività di apprendimento e metodologie didattiche - Teaching method	
Lezioni frontali interattive, esercitazioni guidate in laboratorio, esercitazioni autonome	Interactive front lectures, guided and self-learning laboratory sessions
Metodi e criteri di accertamento del profitto - Assessment method	
Prova scritta in laboratorio, progetto didattico e prova orale	The assessment is divided into three stages: a programming exam taken at the PC, a small project consisting in designing a given LAN, and an oral exam.