

# Software Librea

-

## **SISTEMA ERAGILEAK eta TCP/IP: Oinarrizko kontzeptuak: prozesuak, memoriaren kudeaketa eta dispositiboen zein fitxategien kudeaketa**

**IÑAKI ALEGRIA - ROBERTO CORTINAS**



# AURKIBIDEA

<b>1. SARRERA.....</b>	<b>3</b>
<b>2. MOTIBAZIOA ETA HELBURUAK.....</b>	<b>3</b>
<b>3. OINARRIZKO KONTZEPTUAK.....</b>	<b>3</b>
3.1 Sarrera.....	3
3.2 Prozesuen adierazpidea eta kudeaketa.....	4
3.3. Prozesadorearen planifikazioa.....	5
3.4. Memoriaren kudeaketa.....	5
3.5. Sarrera/Irteerako gailuen kudeaketa.....	6
3.6. Fitxategi-sistema.....	6
<b>4. BIBLIOGRAFIA.....</b>	<b>8</b>
4.1 Liburuak.....	8
4.2 Webguneak.....	8

# 1. SARRERA

Gaur egungo sistema eragileak oso konplexuak dira: aplikazioen exekuzioa kudeatzeaz eta optimizatzeaz gain, makinaren baliabideei etekina atera eta, batzuetan, bere gabeziak *estali* behar ditu. Gai honetan aztertuko dugu nolakoak diren sistema eragile baten barnekoak. Sistema eragileak kudeatu behar duen baliabide bakoitzeko azpial bat dago: prozesuak, prozesadorea, memoria, sarrera/irteerako gailuak eta fitxategi sistema. Hala ere, gai honetan kontzeptu orokorren aurkezpena baino ez da egiten, azterketa sakonik bilatu gabe, inplementazio errealen konplexutasuna saihestu nahian.

# 2. MOTIBAZIOA ETA HELBURUAK

Sistema eragilea software mota bat da eta, erabiltzaileak zuzenean eskatutakoa eskaintzen ez badu ere (ez du kalkulurik egiten, ez du dokumenturik editatzen ...), softwarerik erabiliena da, baita konplexuenetako bat ere. Hortaz, softwarea aztertuz gero, ezinbesteko urratsa da sistema eragileak ere aztertzea.

Software librearen munduan badaude hainbat sistema eragile inplementaturik, baina beharbada ezagunena *Linux* dugu. *Linux* oso eredu ona da sistema eragileen ezaugarriak aztertzeko garaian. Hala ere, *Linux* aztertzen hasi baino lehen, oso komenigarria da sistema eragileen kontzeptu orokorrak ezagutzea, gerora *Linux*en nola aplikatzen diren ulertu ahal izateko.

# 3. OINARRIZKO KONTZEPTUAK

## 3.1 Sarrera

**Hardwarearen oinarrizko arkitektura:** Hiru elementuk osatzen dute hardwarearen oinarrizko egitura: prozesadorea, memoria nagusia eta sarrera/irteerako gailuak.

- **Prozesadorea** (edo *CPU*): Prozesuen aginduak exekutatzeko gailua. Honen barruan ondorengo osagarriak aurki ditzakegu: Unitate Aritmetiko Logikoa, UALa (*Aritmetic-Logic Unit, ALU*), eragiketak egiteko, aldi baterako datuak gordetzeko erregistroak (erregistro-multzoa), exekuzioaren inguruko erregistro bereziak (*Program Counter* edo *PC*, *Program Status Word* edo *PSW*, *Instruction Register* edo *IR* ...).
- **S/Iko gailuak** (edo *devices*): Aplikazioek sarrerako datuak irakurri edota irteerakoak idatzi behar dituzte. Datu horiek lortzeko edo gordetzeko S/Iko gailuak erabiltzen dira. Esaterako, teklatura sarrerako dispositiboa da, pantaila irteerakoa eta diskoa sarrera eta irteerakoa izan daiteke.
- **Memoria nagusia:** Aurrean aipatutako gailuak oso motelak dira, eta gainera oso ezaugarri desberdinetakoak izan daitezke. Aginduak eta datuak era bateratuan eta azkar atzitzeko memoria erabiltzen da. Beraz, programa bat exekutatzean bere aginduak eta erabiltzen dituen datuak gailuetatik memoriara eraman eta bertan atzitzeko ditu prozesuak.
- **Busak:** Aurreko hiru elementuen artean informazioa trukatzeko lotzen dituen seinale-lerroak (zirkuitu mailan).

**Multiprozesadorea:** Aurreko arkitekturaren prozesadore bat baino gehiago dagoenean, prozesuen exekuzioa arintzeko helburuarekin. Bi motakoak izan daitezke; asimetrikoak eta simetrikoak. Lehenengo kasuan prozesadore batek besteak kudeatzen ditu. Bigarren kasuan lana denen artean banatzen da. Azken kasu honetan SMP (*Symmetric MultiProcessing*) motako sistema daukagu.

**Multikonputagailua:** Konputagailu-sare bat da, aurreko kasuan bezala, prozesadore bat baino gehiago duena. Kasu honetan memoria ez da konpartitua, prozesadore bakoitzak bere memoria baitauka.

**Sistema eragilea** (edo *operating system*): Erabiltzailearen aplikazioak eta makinaren arteko bitartekaria. Erabiltzailearen aplikazioei alegiazko makina eskaintzen die, gailuen berezitasunak ezkatutuz. Aplikazioek alegiazko makina hori API (*Application Programming Interface*) baten bitartez erabiltzen dute.

**Nukleoa** (edo *kernel*): Sistema eragilearen muina, oinarrizko funtzionalitatea ematen duena. Bertan prozesuak eta sistemaren baliabideak kudeatzen dira. Nukleoko kodea exekutatzean *kernel moduan* edo sistema moduan exekututzen da, eta ez dauka murriztapenik. Aplikazioak, berriz, *user moduan* edo *modu babestuan* exekututzen dira eta murriztapenak daude (esaterako ezin dute bere memoria-tartetik kanpo dagoen helbiderik atzitu).

**Direct Memory Access** (edo *DMA*): Hardwarezko modulua da; memoria eta gailuen arteko datutrukaketaren kudeaketaz arduratzen da, prozesadoreari lan horren ardura kenduz. Era honetan transferentzia egiten den bitartean prozesadoreak beste eginkizunetan jarrai dezake.

**Hardware Abstraction Layer** (edo *HAL*): sistema eragilearen modulu orokor bat hardwarea kudeatzeko, sistema eragilearen goiko maileri gailuen berezitasunak ezkatutuz.

**Komando-interpretatzailea, Shell:** Sistema eragileak erabiltzailei eskaintzen dien oinarrizko interfazea.

**Sistema-deia** (edo *system call*): Prozesu baten sistema eragileari egindako eskaera.

**Application Programming Interface** (edo *API*): Sistema eragileak (edo beste *software* batek) aplikazioei eskaintzen dien interfazea.

### 3.2 Prozesuen adierazpidea eta kudeaketa

**Programa:** Zeregin zehatz bat gauzatzeko programatutako agindu-multzoa. Sarrerako datuak prozesatu eta irteerako datuak emango ditu. Programak konpilatuak edo interpretatuak izan daitezke. Lehenengo kasuan agindu-multzoa makina-lengoaian dago idatzita eta zuzenean exekutatu daiteke. Bigarren kasuan, berriz, aginduak goi-mailako lengoaia batean adierazita daude eta exekutatu aurretik itzuli behar dira.

**Prozesua** (edo *process*): Programa baten exekuzio-instantzia, exekuzio horren datuak baneratzten dituena. Programa batek  $n$  prozesu izan dezake aldi berean exekuzioan.

**Haria** (edo *thread*): *prozesu arinak* (*lightweight process*) ere deituak. Prozesuen arteko komunikazioaren kostua eta lan-karga gutxitzeko sortutako kontzeptua. Prozesu batek hari batzuk izan ditzake eta guztiek prozesu horren baliabideak konpartitzen dituzte. Prozesu bereko harien arteko komunikazioa ez du sistema eragileak kudeatu behar. Hariak erabiltzen diren sistemetan prozesadorerako planifikazio-unitatea haria da.

**Exekuzio-egoerak:** Prozesuaren exekutatzeko den bitartean, zenbait egoeretan izan daiteke. Nagusiak ondorengoak dira: hasieratzen (*berria*), prozesadorean noiz sartuko zain (*prest*), prozesadorean bertan (*exekuzioan*) edo gertaera baten zain (*blokeatuta*).

**Liburutegia:** Programa baten modulu bat, oro har funtzionalitate multzo bati lotuta. Programak modulutan banatzearen abantaila nagusiak bi dira: (a) prozesuen artean konpartitzeko aukera. Behin baino ez da diskoan izango,  $n$  prozesu erabili arren. Horrez

gain, liburutegiaren eguneratzea errazago izango da. (b) Memoriako espazioaren kudeaketa hobetzea (ikus hurrengo puntua).

**Estekatze dinamikorako liburutegiak** (edo *Dinamic Link Library*): *DLL* eta *run-time library* ere deitua. Behar direnean (eta ez hasieratik) memorian dinamikoki kargatzen diren liburutegiak dira. Era honetan memoria aurrezten da, baina, horrez gain, hainbat prozesuk konparti dezakete memorian kargatuta dagoen liburutegia (behin baino ez dago memorian). *Unixen* *.so* luzapenarekin aurki ditzakegu */usr/lib* katalogoan.

**Multiprogramatua**: Aldi berean prozesu bat baino gehiago exekutatzeko aukera izatea. Era honetan sistemaren baliabideak hobeto aprobetxatuko dira. Hala ere, kudeaketaren konplexutasuna handiagoa izango da eta, ondorioz, kudeaketa horrek kontsumitzen duen denbora ere. Kontrolmekanismoak jarri ezean kontrako efektua sortu daiteke multiprogramazioan, sistemaren eraginkortasuna jaitsiz.

**Testuinguru-aldaketa** (edo *context switch*): Prozesu baten exekuzioa etetea, dagokion testuinguru gordez, eta prozesadoreari egokitzen zaion hurrengo prozesuaren testuinguru bertan ezartzea. Horren arduraduna *dispatcher* izeneko funtzioa da.

**Dispatcher**: Multiprogramazioa bideratzeko oinarrizko mekanismoa. Prozesuen artean prozesadorearen banatzeaz arduratzen da.

**Etena** (edo *interrupt*): Hardwarezko mekanismoa, exekutatzen ari den prozesuaren exekuzioa al-di batez moztu eta ekintza bat gauzatzeko (eten bakoitzari dagokion arreta-errutina, agindumultzoa exekutatuz). Eten mota batzuk daude.

**Desbiderapena** (edo *trap*): Programek zuzenean eragindako etena, errore kontrolatuak adierazteko gehienetan.

### 3.3. Prozesadorearen planifikazioa

**Scheduler**: Prozesadorearen planifikaziorako funtzio logikoa (epe laburreko planifikazioa). Prozesadorean prozesu bat sartu behar denean, *prest* egoeran daudenen artean aukeratu behar duena.

**Planifikazio-politika**: Prozesuen artean prozesadorea banatzeko politika. Horietako batzuk dira FCFS, SJF, lehentasuna eta txandakatzea.

**Lehentasun dinamikoa**: Prozesadorearen planifikazio-politikaren aldagai bat. Lehentasuna irizpide da prozesadorean sartzeko, baina prozesuen lehentasuna aldatu daiteke exekuzioan zehar. Era honetan prozesuen arteko oreka bilatzen da.

**Kanporatzailea** (edo *preemptive*): Prozesadorearen planifikazioan, exekuzioan dagoen prozesua kanporatzeko aukera izatea (amaitzen edo blokeatzen den arte itxaron behar izan gabe). Bi motakoa izan daiteke: gertaeren bidezkoa edo denbora bidezkoa.

### 3.4. Memoriaren kudeaketa

**Memoria**: Prozesadoreak erabiltzen duen informaziorako oinarrizko biltegia. Memoria *fisikoa* da hardwareak eskaintzen duena eta *logikoa* prozesuek erabil dezaketena.

Memoriaren egitura

**Esleipen-unitatea**: Memoria fisikoa gelaxka izeneko unitatetan banatzen da. Hauen tamaina minimoa bytea delarik,  $1$  byteko gelaxkak erabiltzen dituen  $1$  GBko memoria batean  $2^{30}$  gelaxka<sup>1</sup> daude. Beste aukera bat esleipen-unitatea *hitza* izatea da. kasu honetan erregistroen tamainakoa da (32 edo 64 bitekoa).

<sup>1</sup>  $1 \text{ G} = 2^{10} \text{ M} = 2^{10}(2^{10}) \text{ K} = 2^{10}(2^{10})(2^{10})$ , beraz  $1 \text{ GB} = 1 \text{ G byte} = 2^{30} \text{ byte}$

**Helbidea:** Memoria baten esleipen-unitateak helbideratu ahal izateko bakoitzari esleitzen zaion identifikadorea. Identifikadore hori *helbidea* da. Aurreko kasuan  $2^{30}$  gelaxka helbideratu ahal izateko 30 biteko helbideak<sup>2</sup> bit beharko ditugu. Helbidearen tamaina adierazten du zein den helbideratu daitekeen memoriaren tamaina, *helbideratze-tartea* alegia.

**Helbide erreala:** Gelaxka baten helbide fisikoa.

**Helbide logikoa:** Prozesuari eskainitako memoriarekiko helbide erlatiboa.

**Orrikapena** (edo *paging*): Sistema eragileak memoria eta disko artean trukatzeko duen unitatea.

**Memory Management Unit** (edo *MMU*): Hardwarezko gailua, helbide logikoa helbide fisikora itzultzeko.

**Swapping:** Memoriaren tamainaren muga gainditzeko mekanismoa, bigarren mailako memoria bat (adibidez, diskoa) erabiliz.

**Alegiazko memoria** (edo *virtual memory*): Memoria handitzeko mekanismoa. Prozesuak guztiz memorian kargatu gabe exekutatzeko aukera ematen du. *Swapping* teknikaren beharra dauka.

### 3.5. Sarrera/Irteerako gailuen kudeaketa

**Sarrera/Irteera, S/I** (edo *Input/Output, I/O*): Sarrerako datuak eta irteerako datuak lantzen dituen hardwarearen atala, baita bere tratamendua sistema eragilean ere.

**Kontrolagailua:** Gailuak izan dezakeen hardwarezko kontroladorea, gailua programatzeko. CPUarekin lotzen duen S/Iko *bus*arekin dago konektatuta.

**Kudeatzailea, kontrolatzailea** (edo *driver*): Sistema eragileari gailuaren ezaugarriak eta erabiltzeko era adierazten dioen kode-zatia edo errutina-multzoa.

**Buffering:** *Buffer* izeneko memoriako datu-biltegia erabiltzen duen teknika. Bi elementuen arteko komunikazioa bideratzeko erabiltzen duena;  $A$  tik  $B$  ri informazioa bidaltzeko,  $A$  k komunikaziokanalaren bufferrean idazten du. Bertatik  $B$  k irakurri ahal izango du. Teknika honi esker, besteak beste, abiadura desberdina daukaten elementuak komunikatu ahal izango dira, esaterako, prozesuak (prozesadorean exekutatuta) eta S/Iko gailuak.

**Spooling:** Gailu batzuekin, inprimagailuak esaterako, komunikazioa bideratzeko teknika bat. Bi prozesuk aldi berean inprimagailua erabili nahi baldin badute arazoak sor daitezke lanak nahasten badira. Egoera hau ekiditeko lan bakoitza buffer independentetan, *spool* izenekoak, metatzen dira (*Linux*en fitxategiak dira diskoan) eta banan-banan tratatzen dira.

**Diskoa:** Informazioa modu iraunkorrean metatzeko erarik erabiliarena. Hainbat motakoak daude, baina gaur egun disko magnetikoak dira sistema eragileak gehien erabiltzen dituenak.

- **MBR** (edo *Master Boot Record*): Diskoaren lehenengo sektore fisikoa, diskoaren egituraren inguruko informazioa daukana (partizio-taula).
- **Partizioa:** Disko bat zati logikoetan banatu daiteke, partizioak, eta bakoitzari helburu (sistema edo datuak) eta ezaugarri desberdinak esleitu (tamaina, fitxategi-sistema

<sup>2</sup> 2 kodeketa bitarrez,  $2^{30}$  aukera adierazteko  $\log_2 2^{30} = 30$

mota . . . ). Disko batek dauzkan partizioen inguruko informazioa diskoaren hasieran dagoen partizio-taulan aurki daiteke (ik. MBR).

- **S/Iko busak:** gailuak konektatzeko busak oso bestelakoak izan daitezke. Esaterako, diskoak konektatzeko, besteak beste, ondorengo aukerak aurki ditzakegu, (E)IDE (*Enhanced integrated drive electronics*), ATA (*advanced technology attachment*), SATA (*serial ATA*) edo USB (*universal serial bus*). Haien arteko desberdintasunak abiaduran, kostuan eta beste ezaugarri batzuetan daude.

### 3.6. Fitxategi-sistema

**Fitxategi-sistema** (edo *file system*): Bigarren mailako gailuetan informazioa (fitxategiak) antolatzeko sistema.

**Erregistroa:** Erabiltzaileren aplikazioak erabiltzen duen diskoko unitate logikoa. Erregistro honen egitura eta tamaina aldatzen da aplikazioaren arabera. Esaterako *txt* (testu) motako fitxategietan karakterea dugu erregistro logikoa.

**Fitxategia** (edo *file*): Hainbat ezaugarri (izena, atzipen-eskubideak . . . ) duen informazio-unitatea logikoa. Datu-fitxategien kasuan, erregistro logiko multzo bat izango da.

**Katalogoa** (edo *directory*): Informazioa (fitxategitan egituratuta) antolatzeko modua, fitxategiak multzokatuz katalogoaren barruan. Hasierako fitxategi-sistemetan katalogo bakarra zegoen (*errokatalogoa*), baina gaur egun edozein fitxategi-sistemak katalogo bat beste baten barruan izateko aukera eskaintzen du, zuhaitz-egitura bat sortuz.

**i-nodoa:** Fitxategi-sistema mota batzuetan fitxategi baten atributuak (tamaina, baimenak, kokapena . . . ) gordetzeko unitatea.

**ext3 fitxategi-sistema:** *Unixen* eta *Linuxen* ohiko fitxategi-sistema. *ext2* fitxategi-sistema oinordekoa da eta hobekuntza nabarmenak ditu, besteak beste *journaling* izenekoa.

**Journaling:** Sistema transakzionala fitxategi-sisteman. Fitxategien antolaketaren inguruko aldaketak gordetzen dira egin aurretik (*egunkari* edo *journal*, batean). Era honetan, aldaketak egiten diren bitartean errore larri bat sortuz gero, sistema eragileak informazioaren koherentzia berreskuratzeko aukera izango du.

**FAT fitxategi-sistema:** *Microsofteko* fitxategi-sistema bat, FAT taula erabiltzen duena. Oso hedatuta dagoen arren, oso mugatua da (fitxategien tamaina maximoa, segurtasun-eza . . . ).

**NTFS fitxategi-sistema** (edo *New Technology File System*): *Microsofteko* fitxategi-sistema berria, FAT fitxategi-sistema ordezkatzeko. Besteak beste, fitxategien tamaina askoz handiago izan daiteke, partizio handiak hobeto kudea daitezke, segurtasuna gehitzen du eta eraginkorragoa da.

**RAID:** edo *redundant array of inexpensive (edo independent) disks*. Diskoaren eraginkortasuna eta erroreari aurre egiteko gaitasuna handitzeko sistema (softwarezkoa ala hardwarezkoa izan daiteke). RAID sistemetan atzipenak disko multzo baten artean bana daitezke edota erredundantzia erabili. Era honetan 7 maila osatzen dira, bakoitza bere ezaugarriekin.

## 4.BIBLIOGRAFIA

### 4.1. Liburuak

- [1] W. Stallings. Sistema Eragileak: Barnekoak eta diseinuko printzipioak. UPV/EHU, 2008.
- [2] A. S. Tanenbaum. Konputagailuen antolaketa egituratua. Euskal Herriko Unibertsitatea, 2008.
- [3] A. Silberschatz, P. B. Galvin, and G. Gagne. Operating System Concepts Essentials. John Wiley & Sons, Inc., 2010.
- [4] S. Sanchez Prieto. Sistemas Operativos. UNIVERSIDAD DE ALCALA DE HENARES, 2nd edition, 2005.
- [5] C. Rodríguez, I. Alegría, J. González, A. Lafuente, and E. Lazkano. Sistema eragileen deskribapen funtzionala. Euskal Herriko Unibertsitatea, 1995.

### 4.2 Webguneak

- [6] Wikipedia: Operating system. [http://en.wikipedia.org/wiki/Operating\\_system](http://en.wikipedia.org/wiki/Operating_system)
- [7] P. Brinch Hansen, editor. *Classic operating systems: from batch processing to distributed systems*. Springer-Verlag New York, Inc., 2001. <http://books.google.com/?id=-PDPBvIPYBkC&lpg=PPI&pg=PPI#v=onepage&q>