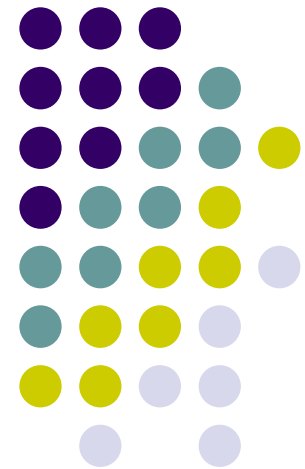
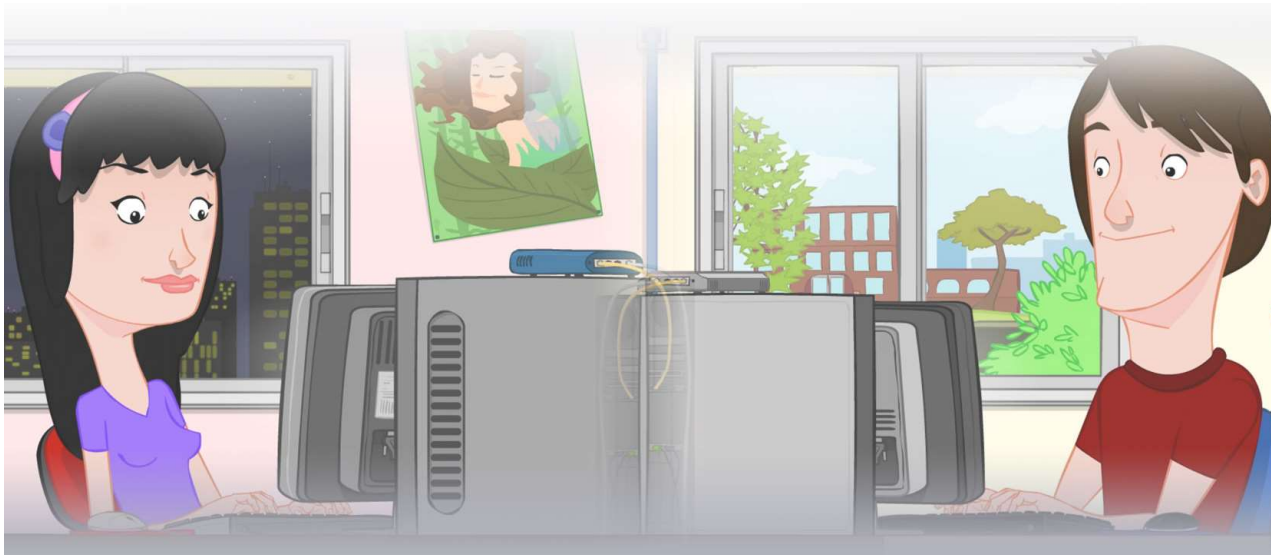


1. Blokea: Sistema Eragileak

Informazio-Sistemen Arkitektura

Telekomunikazio Teknologiaren Ingeniaritzako Gradua (3. Maila)

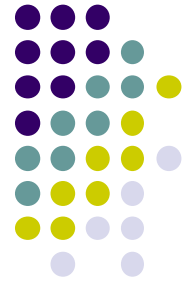


TELEK:O
UPV/EHU Bilbao



1. Blokea - Edukiak

1. Ordenagailuen Arkitekturako kontzeptuak
2. Sistema Eragileak. Sarrera.
3. Prozesuak
 - Prozesuak eta hariak
 - Prozesuen arteko komunikazioa
 - Komunikazio eta Sinkronizazio Mekanismoak
 - Planifikazioa
4. Sarrera/Irteera
5. Memoriaren Kudeaketa
 - Memoria birtuala edo alegiazko memoria
 - Orrikapena eta Segmentazioa
 - Ordezkapeneko algoritmoak
6. Fitxategi Sistema
 - Fitxategiak eta direktorioak
 - Fitxategi Sistemaren antolaketa



2. Gaia – Edukiak

SISTEMA ERAGILEAK. SARRERA

- Zer da Sistema Eragile bat?
- Sistema Eragilea abiarazten
- Osagaiak eta egiturak
- Sistema Eragilearen zerbitzuak
- Zerbitzuen aktibazioa

2. SISTEMA ERAGILEAK.

SARRERA



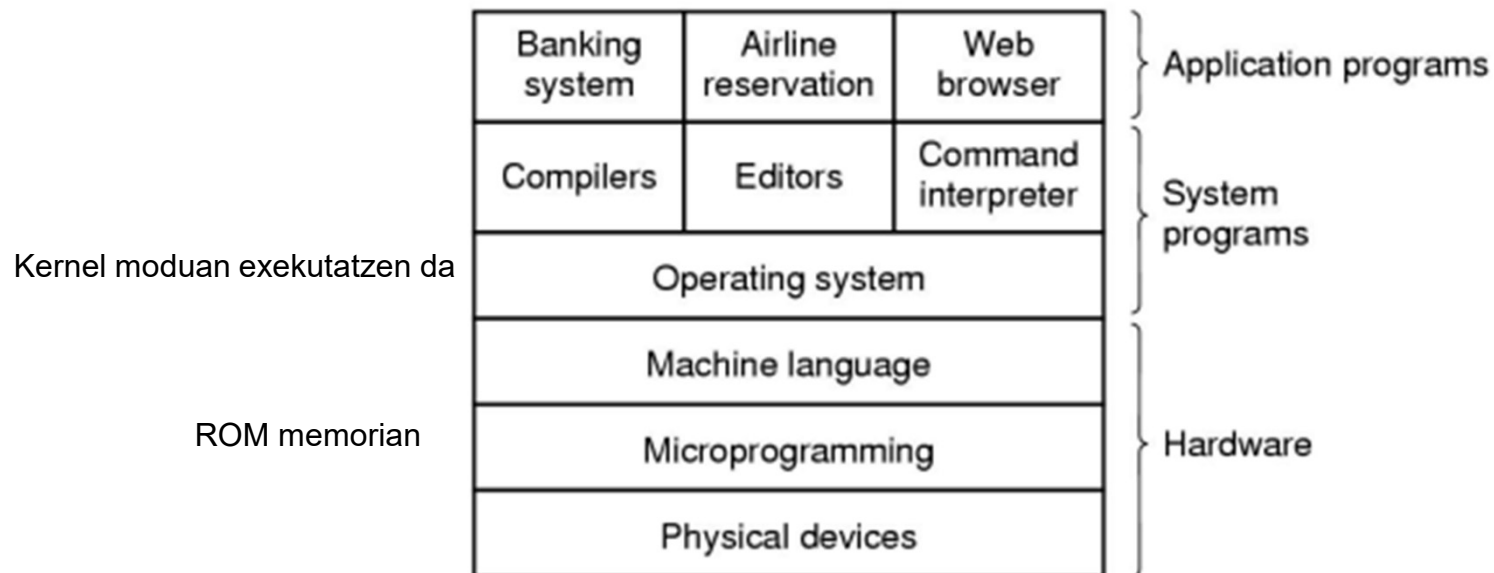
- 2.1 Zer da Sistema Eragile bat?
 - Makina biluzia:
 - Konputagailua: Prozesadorea(k)+RAM+Erlojuak+Terminalak+Diskoak+Komunikazio-interfazeak+... = Sistema konplexu samarra programatzaileentzako → Erabiltzaileentzat.
 - pc-a programen hasieran jarri behar da abiatzerakoan...
 - Makina “jantzi” beharra dago:
 - Erabiltzailearen (programatzailearen) eta HWren arteko interfazea
 - Makina desberdinak modu berdinean erabili
 - Programak exekutatu eta erabiltzaileari arazoak erraztu
 - Baliabideak modu egokian/erosoan erabili

2. SISTEMA ERAGILEAK.

SARRERA



- 2.1 Zer da Sistema Eragile bat?



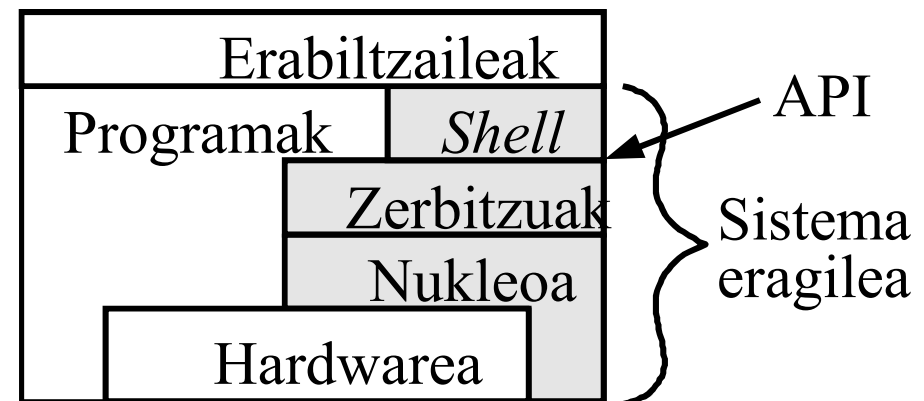
- Bi ikuspuntu:
 - Baliabide kudeatzailea
 - Erabiltzaile interfazea

2. SISTEMA ERAGILEAK.

SARRERA



- 2.1 Zer da Sistema Eragile bat?
 - Maila kontzeptualak:
 - **Kernel:** Nukleoak HW baliabideak eta sistemaren oinarrizko funtzionalitateak kudeatzen ditu.
 - **Zerbitzuak.** API Aplikazioak Programatzeko Interfazea edo deiak.
 - **Shell:** Komandoak jaso, interpretatu eta exekutatu egiten ditu.



- Erabiltzaile(ak)
 - Kontu bakoitzak bere uid, profil eta eskubideekin
 - Supererabiltzailea, pribilegiaduna
 - Erabiltzaile-taldeak euren gid eta eskubideekin

2. SISTEMA ERAGILEAK.

SARRERA



- 2.1 Zer da Sistema Eragile bat?
 - **Baliabide kudeatzaile** moduan:
 - Azpitik gorako ikuspegia ↑.
 - Prozesuei HW eta SW baliabideak esleitu/berreskuratu
 - Baliabide erabileran borroka ekin
 - RAM kopuruarekiko, PUZ motarekiko, HD... erabilera “independentea”.
 - Babesa
 - Informazioaren konfidentzialtasuna
 - Lanak elkarren artean ez nahastatzea
 - Baliabideen kontabilitatea eta monitorizatzea

2. SISTEMA ERAGILEAK.

SARRERA



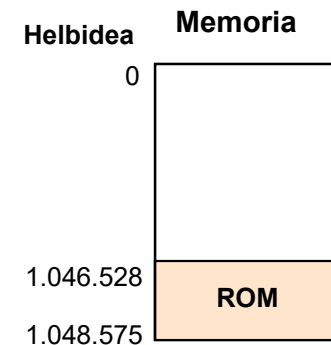
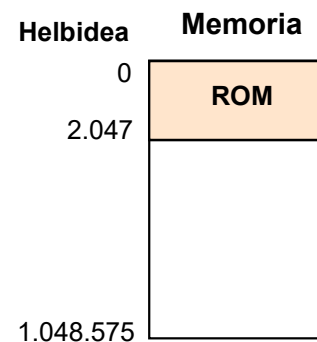
- 2.1 Zer da Sistema Eragile bat?
 - Erabiltzaile interfazea. “**Makina hedatu**” moduan:
 - Goitik beherako ikuspegia ↓ .
 - Sistema deiek HW xehetasunak estaltzen dizkiote programatzaileari: Makina hedatua.
 - Shell edo komando-interpretatzailea, begizta infinituan.
 - Programak errazagoak dira, seguruagoak.
 - Interfaze grafikoa
 - Zerbitzuak: Agindu-fitxategiak edo scriptak (komandoak + exekuzioko fluxuaren kontrola)
 - Programak exekutatu: Programak abiarazi, prozesuak sortu, exekutatu, komunikatu, sinkronizatu, amaitu.
 - S/I aginduak: Irakurri/Idatzi/Egoera. Abstrakzioa.
 - Fitxategiak: Sortu, irakurri, idatzi, ezabatu...
 - HW eta behe-mailako SW erroreak antzeman eta tratatu.

2. SISTEMA ERAGILEAK.

SARRERA



- 2.2 Sistema Eragilea abiarazten
 - *Reset/Boot* egiterakoan PUZaren erregistroetan balio lehenetsiak kargatzen dira
 - pc ← Bootloader-en helbidea
 - Sistemaren Bootloader kargadorea exekutatu
 - Sistemaren Testa
 - SEaren boot sektorea ekarri



- Bootloader-a (HW-ekiko independentea) != SEaren kargadorea (HW-ekiko dependentea)
- ROMean S/I oinarritzko SW ere badago: BIOS/UEFI

2. SISTEMA ERAGILEAK.

SARRERA



- 2.2 Sistema Eragilea abiarazten

Bootloader-aren
menpe

{ HW Testa
SE abio sektorea abiarazi

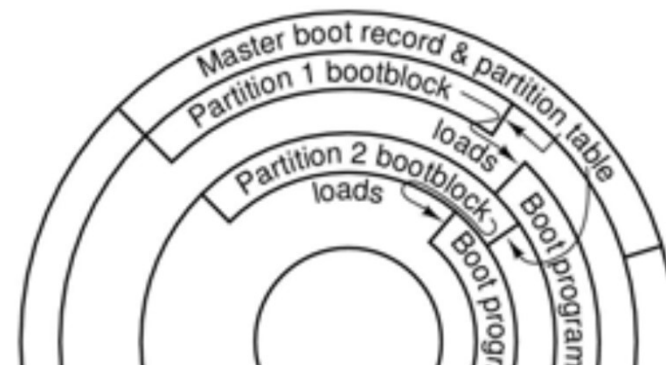
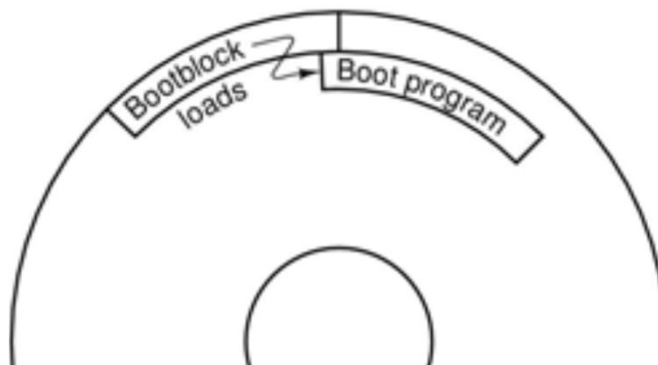
SEaren abio programaren
menpe

{ SEren osagaiak RAMean kargatu

SE osagai egoiliarren
kontrolpean

{ Fitxategi Sistemaren Testa
Barneko datu-egiturak sortu
SE egoiliarraren karga osoa
Login prozesuak sortu

SEaren funtzionamendu arrunta hasi

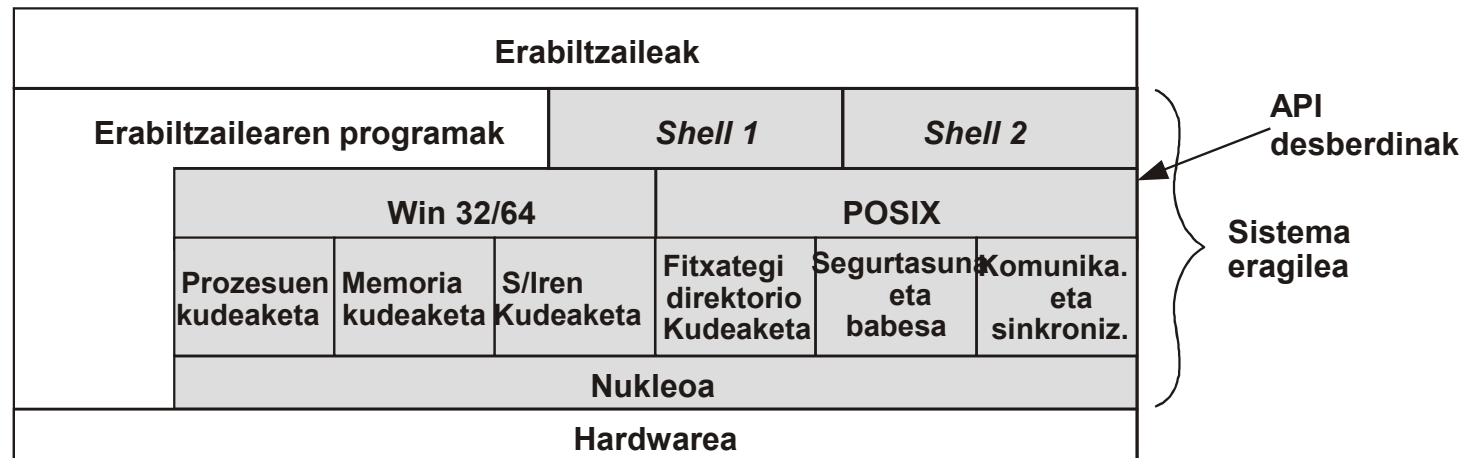


2. SISTEMA ERAGILEAK.

SARRERA



- 2.3 Osagaiak eta egiturak
- 3 Maila kontzeptual: Kernel, Zerbitzuak eta Shell.



- Kernel: HW, PUZa kudeatu, IRQak, memoria..
- Zerbitzuak: Prozesuak, memoria, S/I, fitxategiak eta direktorioak kudeatzeko, prozesuak komunikatu eta sinkronizatzeko, segurtasunerako eta babeserako
- Makina berean Sistema eragile bat edo gehiago, sistema eragile bakoitzak bere APIa izango du (POSIX, Win32): Makina hedatu desberdinak

2. SISTEMA ERAGILEAK.

SARRERA



- 2.3 Osagaiak eta egiturak
 - 3 paradigma sistema eragileen diseinuan:
 - 2.3.1 Sistema eragile monolitikoak
 - 2.3.2 Sistema eragile egituratuak
 - Geruzetan
 - Bezero-zerbitzaria
 - 2.3.3 Sistema eragile banatuak

2. SISTEMA ERAGILEAK.

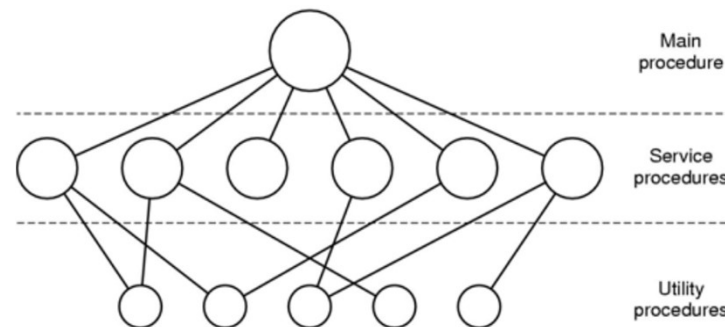
SARRERA



- 2.3 Osagaiak eta egiturak

- 2.3.1 Sistema eragile **monolitikoak**:

- Egitura argirik ez
- Osagai guztiak programa bakar batean. Kernelean.
 - Helbideratze espazio bakarra
 - Osagai guztiek partekatzen dituzte datu egiturak eta funtzioak -> komunikazioa eta sistemako baliabideetara sarbidea errazten ditu
 - SE guztia batera konpilatu/estekatu beharra



Sistema deiak egikaritzeko

2. SISTEMA ERAGILEAK.

SARRERA



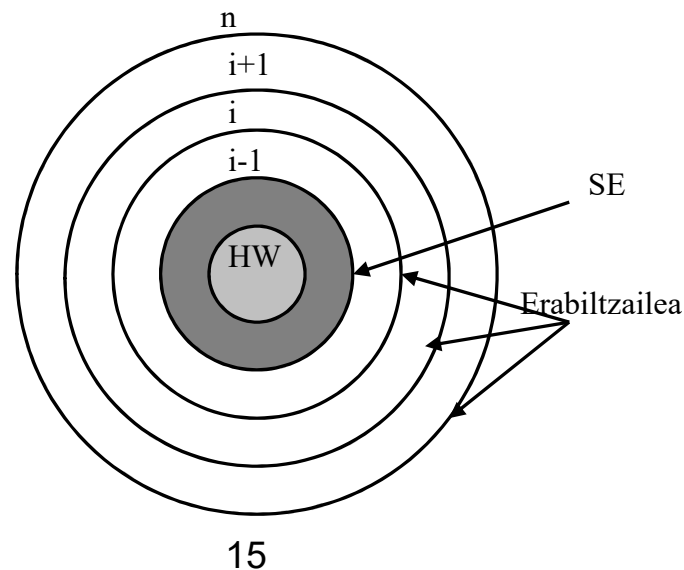
- 2.3 Osagaiak eta egiturak
 - 2.3.2 Sistema eragile **egituratuak**:
 - Sistema eragilearen funtzioak modulu edo geruza independenteetan banatzen dira
 - Komunikazioa eta sistemako baliabideetara sarbidea definituriko interfazeen bidez
 - Modulu bakarrak exekutatzen du kernela: mikrokernel
 - Malgutasuna, bloke txikiak
 - Helbideratze espazio desberdinak
 - Babesa
 - Bi mota
 - Geruzetan
 - Bezero-zerbitzari

2. SISTEMA ERAGILEAK.

SARRERA



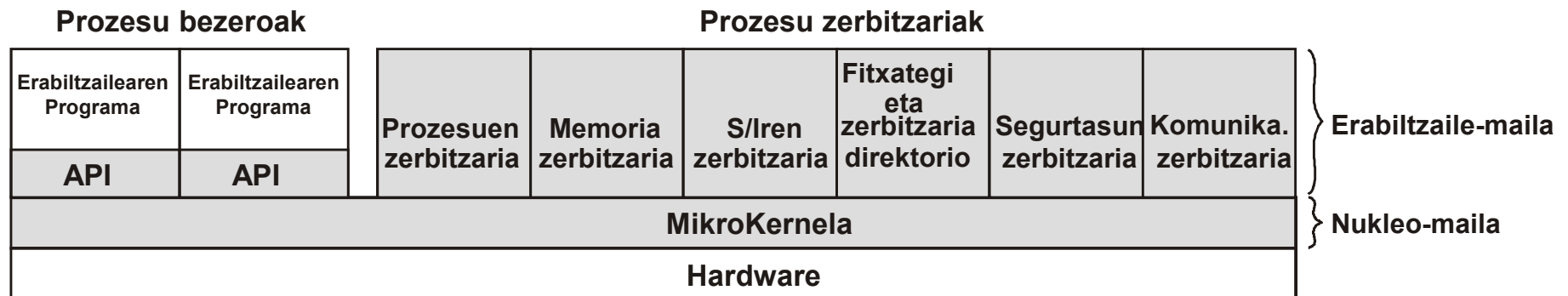
- 2.3 Osagaiak eta egiturak
 - 2.3.2 Sistema eragile **egituratuak**:
 - **Geruzetan** oinarritutako sistema eragileak
 - Geruza desberdinetan zerbitzu desberdinak
 - Barruko eraztunek pribilegio handiagoak dituzte
 - i geruzako prozesuek goragoko prozesuak kontrola ditzakete
 - $i+1$ geruzako prozesuak i geruzako prozesuak erabiltzeko TRAP agindua erabiliko dute



2. SISTEMA ERAGILEAK. SARRERA



- 2.3 Osagaiak eta egiturak
 - 2.3.2 Sistema eragile **egituratuak**:
 - **Bezero-zerbitzari** egituratuak
 - Sistema eragilearen funtzioak bezero-zerbitzari egitura batean banatzen dira
 - Bezeroak: Zerbitzu jakin batzuk eskatzen dituzte
 - Zerbitzariak: Zerbitzu jakin bakoitza zerbitzari batek ematen du
 - Komunikazioa eta sistemako baliabideetara sarbiderako komunikazio baliabideen beharra -> zailtasuna

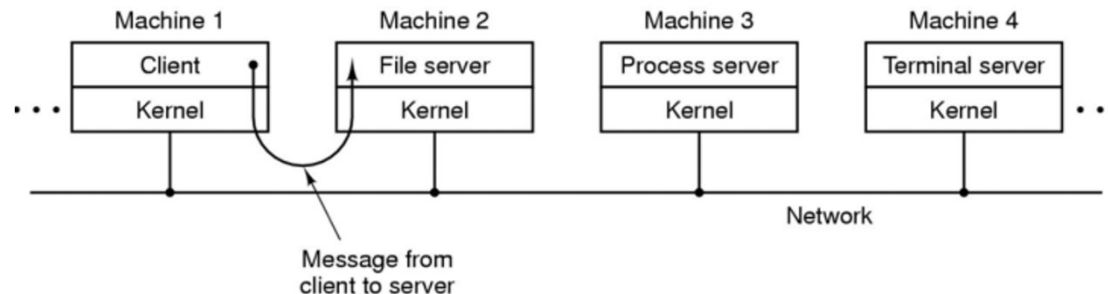


2. SISTEMA ERAGILEAK.

SARRERA



- 2.3 Osagaiak eta egiturak
 - Sistema eragile **banatuak**
 - Aurreko bezero-zerbitzari egitura makina askotara hedatuta
 - Sistemaren baliabideak makina desberdinetan banatuta dauden kasuetarako
 - Prozesuak sareko nodo desberdinetan exekutatu daitezke
 - Komunikazioa eta sistemako baliabideetara sarbidea sare bitarteko mezuen elkartruke bidez ematen da -> zailtasuna+gainzama

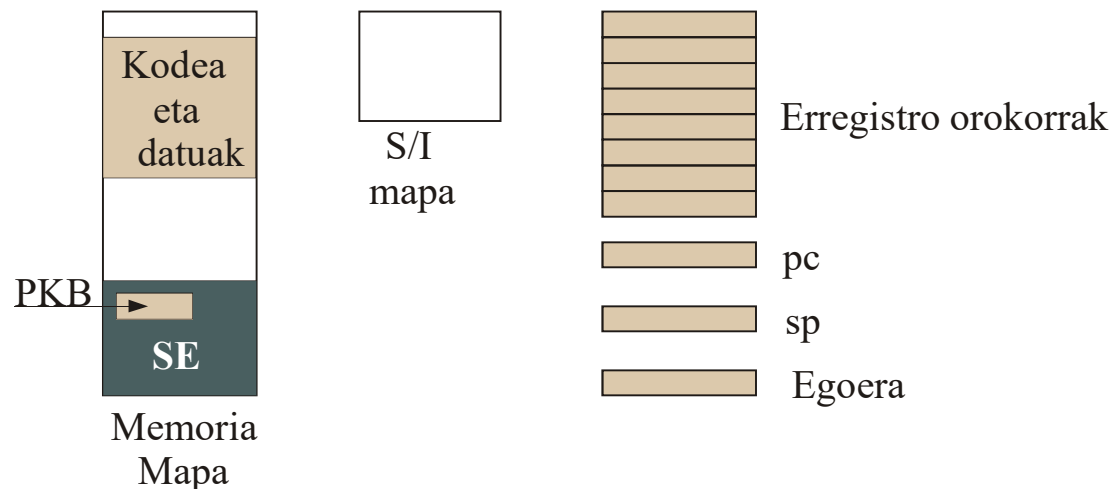


2. SISTEMA ERAGILEAK.

SARRERA



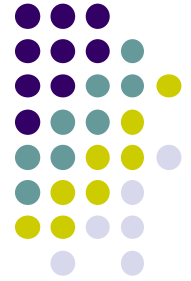
- 2.4 Sistema Eragilearen zerbitzuak
- **Prozesuen kudeaketa**
 - SE-aren oinarrizko elementua
 - Prozesua(k) = programa bat exekuzioan.
 - Prozesu baten elementuak:



- Prozesu bakoitzaren egoera: PKB Prozesua Kontrolatzeko Blokean.
- SEak Prozesuen Taulan PKB guztiak gordetzen ditu

2. SISTEMA ERAGILEAK.

SARRERA



- 2.4 Sistema Eragilearen zerbitzuak
- **Prozesuen kudeaketa**
 - Prozesuak sortzeko zerbitzuak: `fork()`
 - Abian dagoen prozesua aldatzeko: `exec()`
 - Amaitzeko: `exit()`
 - Sinkronizatzeko: `wait()`, `waitpid()`
 - Identifikatzeko: `getpid()`, `getgid()`
 - Seinaleak kudeatzeko: `signal()`, `kill()`, `pause()`, `alarm()`

2. SISTEMA ERAGILEAK.

SARRERA



- 2.4 Sistema Eragilearen zerbitzuak
- **Memoriaren kudeaketa**
 - Libre dagoen eta esleituta dagoen memoriaren mapa izan.
 - Prozesu bakoitzari esleitu zaion memoria kontrolatu.
 - Memoria libre dagoenean, erabaki zein prozesuri eman eta non kokatu.
 - Memoria esleitu eta askatu, behar denean.
 - Memoria birtualaren kudeaketa.
 - Prozesuei memoria partekatzeko aukera eskaini.

2. SISTEMA ERAGILEAK.

SARRERA



- 2.4 Sistema Eragilearen zerbitzuak
- **S/Iren kudeaketa**
 - Kontrolatzaile generikoak, gailu-mota bakoitzeko bana. (SE-ren parte) *Fuck NVIDIA all my homies hate NVIDIA*
 - Kontrolatzaile espezifikoa gailu zehatz bakoitzerako. (Hornitzailearena)
 - Memoria eta gailuen arteko datu-transferentziaren optimizazioa.
 - Ondoko ardurak
 - Eskariak kontrolagailuaren formatura itzuli.
 - Memoria eta kontrolagailuaren artean informazioa kopiatu.
 - DMA bidezko eragiketak zaindu.
 - Terminala, teklatua, sagua etab.
 - Gailuetara egiten diren eskariak planifikatu.
 - S/I Gailu-motak: datuak ze eratan partekatzen dira memoria eta gailuen artean, adib.: karakteretan (teklatua), bloketan (HDD), sekuentzian (sare gailuak)...

2. SISTEMA ERAGILEAK.

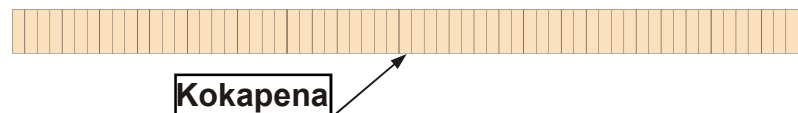
SARRERA



- 2.4 Sistema Eragilearen zerbitzuak
- **Fitxategien eta direktorioen kudeaketa**
 - Ikuspegi logiko eta fisikoak elkarlotu.



Fitxategi baten Ikuspegi logikoa

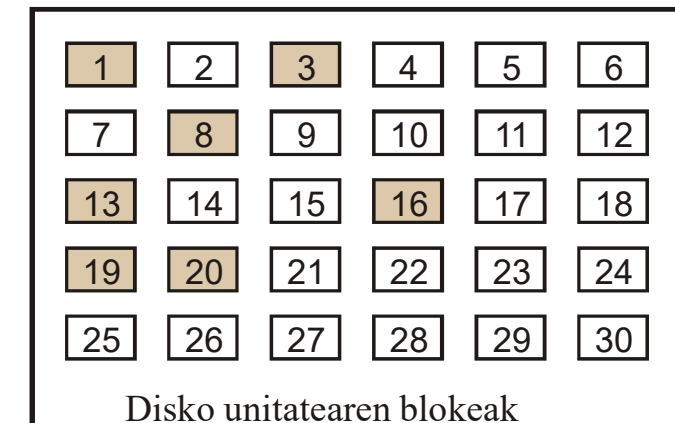


A fitxategia

Blokeak: 13

Ordenean 20

1
8
3
16
19



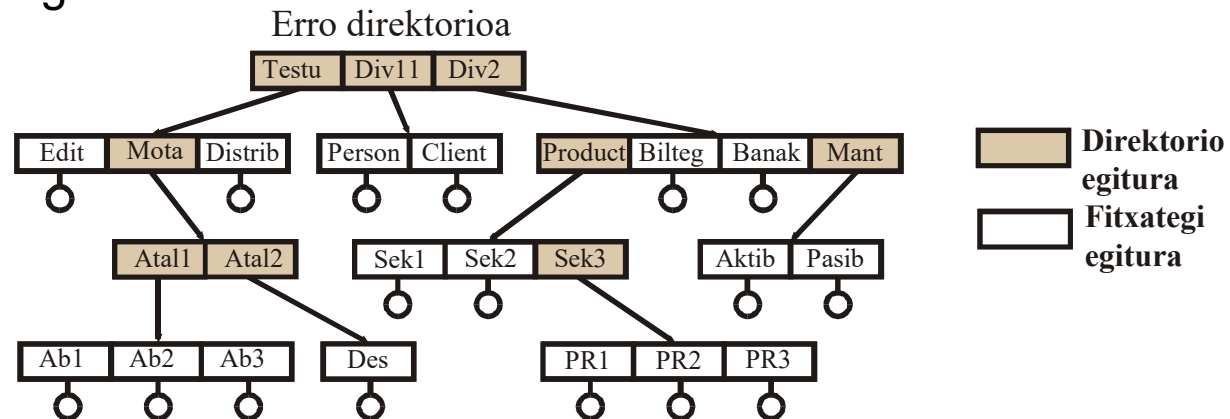
Diskoan fitxategiak fragmentatuta egon daitezke!

Era horrez gain direktorioaren kokapenarekiko ez dago ordenik

2. SISTEMA ERAGILEAK. SARRERA



- 2.4 Sistema Eragilearen zerbitzuak
- **Fitxategien eta direktorioen kudeaketa**
 - Ikuspegi logiko eta fisikoak elkarlotu.
 - Ikuspegi logikoa:



- Ikuspegi fisikoa:
 - Fitxategien izenak dagozkien egitura fisikoekin elkarlotzen dituzten informazio-egiturak

2. SISTEMA ERAGILEAK.

SARRERA



- 2.4 Sistema Eragilearen zerbitzuak
- **Fitxategien kudeaketa**
 - Fitxategiak sortzeko: `creat()`, `mknod()`
 - Irekitzeko, ixteko: `open()`, `close()`
 - Atzitzeko: `read()`, `write()`, `lseek()`
 - Egoera jakiteko: `stat()`, `fstat()`
 - Deskribatzaileak bikoizteko: `dup()`
 - Hodiak sortzeko: `pipe()`, `mkfifo()`
 - Izena aldatzeko: `rename()`
 - Kontrolatzeko: `ioctl()`, `fcntl()`, `access()`...

2. SISTEMA ERAGILEAK.

SARRERA



- 2.4 Sistema Eragilearen zerbitzuak
- **Direktorioen kudeaketa**
 - Direktorioak sortzeko: `mkdir()`, `rmdir()`
 - Estekak sortzeko/ezabatzeke: `link()`, `unlink()`
 - Fitxategi sistemak muntatzeko: `mount()`, `unmount()`
 - Cachea iraultzeko: `sync()`
 - Direktorioa aldatzeko: `chdir()`, `chroot()`

2. SISTEMA ERAGILEAK.

SARRERA



- 2.4 Sistema Eragilearen zerbitzuak
- **Segurtasunaren kudeaketa**
 - Babes bitak aldatzeko: `chmod()`
 - Identitatea jakiteko: `getuid()`, `getgid()`
 - Identitatea jartzeko: `setuid()`, `setgid()`
 - Jabea aldatzeko: `chown()`
 - Babeseko biten maskara jartzeko: `umask()`

2. SISTEMA ERAGILEAK.

SARRERA



- 2.5 Zerbitzuen Aktibazioa
 - SEa lan-eskarien zain dagoen zerbitzari bat da. Lan ezean prozesu nagia exekutatzen du.
Idle
 - Eskarien jatorria:
 - Shell-ak interpretatutako erabiltzaileen komandoen bidez
 - Erabiltzaileen programek sistemara igorritako deiak
 - Periferikoek sortutako etendurak
 - Salbuespen baldintzak eta hardware erroreak
 - Zerbitzuak eskatzeko bidea
 - Dei zuzen baten bidez → segurtasuneko arazoak
 - Etenduren bidez edo TRAP bidez
 - Liburutegiko errutinen bidez:
 - SEra deiak egiteko prestatu beharreko makina aginduak

2. SISTEMA ERAGILEAK.

SARRERA



- 2.5 Zerbitzuen Aktibazioa



2. SISTEMA ERAGILEAK.

SARRERA



- 2.5 Zerbitzuen Aktibazioa
 - Zerbitzuak programazio lengoaia (adib. C) baten sententziak bezala programatzen dira
 - Programatzaile mailako ohiko funtzio baten deia itxura dute
 - Liburutegiko errutinek eskariko parametroak makina aginduetara itzultzen dituzte eta TRAP bat egiten dute nukleo modura pasatzeko
 - TRAPa, etendura guztiak bezala, kernel moduan exekutatuko da SEaren barnean
 - SEak eskatutako zerbitzua bete eta balio bat itzuliko du

- Adibidea:

```
int fork() {    //fork() deia balizko inplementazio bat (adibide gisa)
    int r;
    LOAD R8, FORK_SYSTEM_CALL    //Deiaren zenbakia (mihiztadura-lengoaia)
    TRAP
    LOAD r, R9
    return r;
}
```

2. SISTEMA ERAGILEAK. SARRERA

