



4. Fonaments de sistemes operatius

Miquel Àngel París Peñaranda

Sistemes Informàtics

1r CFGS Desenvolupament d'Aplicacions Multiplataforma



Índex

1. Objectius d'aprenentatge.....	3
2. Components.....	4
2.1. Evolució històrica dels sistemes operatius.....	4
2.2. Elements.....	5
2.3 Estructura.....	6
3. Classificació i tipus.....	9
3.1. Segons la utilització dels recursos.....	9
3.2. Segons la seva interactivitat.....	10
3.3. Segons el nombre d'usuaris.....	11
3.4. Segons el tipus d'aplicació.....	11
3.5. Segons el nombre de processadors.....	11
3.6. Segons la distribució de les tasques del sistema.....	13
4. Funcions.....	15
4.1. Objectius d'un sistema operatiu.....	15
4.2. Funcions.....	16

1. Objectius d'aprenentatge.

1. Conèixer els components.
2. Aprendre la classificació i els tipus.
3. Diferenciar les funcions.
4. Aplicar els avantatges dels sistemes de virtualització.
5. Instal·lar Windows 11 i Ubuntu Desktop.

2. Components.

Els sistemes operatius han evolucionat tant com els mateixos sistemes informàtics. S'han anat adaptant tant als nous components físics que el formen com a les necessitats dels usuaris.

En aquest apartat veurem com s'ha produït aquesta evolució, quins components podem trobar en un sistema operatiu actual i quines són les diferents maneres d'estructurar-los per formar un tot.

2.1. Evolució històrica dels sistemes operatius

L'evolució dels sistemes operatius se sol dividir en generacions, corresponents a diferents fites tecnològiques. Així, tenim:

- **1. generació (1945-1955):** els primers ordinadors estaven basats en **tubs de buit**. **No existien els sistemes operatius**. Es programava directament en codi màquina, generalment a través de panells de cablejat combinables de diferents maneres. A finals de 1950, les targetes perforades (unes cartolines foradades que codificaven instruccions i dades) substitueixen els panells endollables.
- **2. generació (1955-1965):** apareixen els transistors, que permeten construir ordinadors més petits i potents. Els programes es processaven en lots és a dir, quan es carregava un procés aquest tenia a la seva disposició tots els recursos de la màquina fins a finalitzar la tasca- i hi havia un programa de control que interpretava les targetes. Aquest programa de control constitueix un antecedent dels actuals intèrprets de comandes. La programació es duia a terme en targetes perforades (assemblador o FORTRAN).
- **3. generació (1965-1980):** els circuits integrats irrompen a l'electrònica, aconseguint de nou fabricar ordinadors més petits, potents, eficients i econòmics. Apareixen així mateix els sistemes operatius de temps compartit i la multiprogramació (gràcies a la qual un ordinador és capaç de fer més d'una cosa, encara que no simultàniament). Sorgeixen llenguatges de programació com BASIC o PASCAL

- **4. generació (1980-actualitat):** els circuits integrats es generalitzen a gran escala i apareixen els ordinadors personals. Sorgeixen els sistemes operatius de temps real i la multitasca (que permet a un ordinador fer més d'una tasca de forma simultània). Tenim, a més, equips amb múltiples processadors, sistemes operatius en xarxa, distribuïts, etcètera. Molts dels sistemes operatius que fem servir actualment van aparèixer en aquesta generació: Microsoft Windows, Mac OS i GNU/Linux.

2.2. Elements

Els elements que conformen els sistemes operatius són:

El **nucli**, també anomenat kernel, ubicat als fonaments del sistema operatiu i en contacte directe amb el maquinari. Controla les interrupcions, l'assignació de feines al processador o la comunicació entre els diferents programes. En general, s'encarrega de coordinar la resta d'elements del sistema. Per la seva importància dins del nucli, destacarem la tasca del **planificador**, encarregat d'assignar el temps de processador que correspon a cada procés en el cas dels programes en execució. Encara que hi ha diferents mecanismes de planificació, pràcticament tots assignen un temps màxim a cada procés. Quan aquest procés no es pot acabar en aquest temps assignat, torna a la cua de processos preparats i se li assigna temps a un altre procés diferent.

L'**administrador de memòria** gestiona la memòria principal del sistema (RAM) i l'assigna en porcions als diferents processos segons les seues necessitats. Els sistemes operatius actuals apliquen la tècnica de la **memòria virtual per gestionar la memòria física**. **Aquesta tècnica permet un major aprofitament de la memòria principal**. A més, l'administrador de memòria també gestiona quina part de la informació que requereix el procés es carrega a la memòria principal i quina es queda a la memòria secundària, així com el moviment d'una memòria a una altra (swapping), que es produeix quan la memòria principal és plena i quan un procés necessita més memòria per a la seva execució.

La gestió del **sistema d'entrada i sortida** s'encarrega de presentar l'usuari o de desar les dades independentment del dispositiu d'entrada o sortida que utilitzeu. Així, l'usuari rep/emmagatzema informació en un dispositiu d'una mateixa manera, i és el sistema operatiu el que adapta aquesta petició o aquestes dades al funcionament particular del

dispositiu concret. En molts d'aquests dispositius d'entrada/sortida s'usa un **spool** -o dispositiu d'emmagatzematge massiu-, el qual va emmagatzemant tota la informació que s'anirà enviant al dispositiu físic d'entrada/sortida segons estigui disponible. D'aquesta manera es garanteix que cap procés es quedi bloquejat en espera que el dispositiu d'entrada/sortida estigui disponible.

D'altra banda, tenim l'**administrador de fitxers**, que és el responsable de l'estructura de les dades i programes de cadascun dels usuaris. Vigila la creació, la modificació i l'esborrament de fitxers i carpetes. Una altra funció important del mateix és la gestió dels **permisos i privilegis** dels usuaris sobre els fitxers o carpetes, de manera que només aquells usuaris que estan autoritzats puguin fer-ne ús.

Algunes vegades es considera l'**intèrpret de comandes** un element més dels sistemes operatius, ja que funciona com una interfície entre l'usuari i el sistema, i és l'encarregat d'interpretar i traduir les ordres de l'usuari perquè siguin processades pel sistema.

2.3 Estructura

Monolítica: és la que presentaven els primers sistemes operatius. En realitat, equival a no tenir estructura. Consistia en un sol programa amb rutines que es podien cridar entre si. Aquestes estructures eren molt difícils de mantenir i, gairebé sempre, es fabricaven expressament a la mida del sistema en què s'executarien.

Jeràrquica o per **capes**. Segons els usuaris es van tornar més exigents, els sistemes operatius van incrementar també la seva complexitat i funcionalitat. Per aquests motius, i per facilitar el manteniment dels equips, es va fragmentar el sistema operatiu en parts més petites, amb funcions clarament diferenciades i que oferien una interfície per operar amb les capes o mòduls adjacents.

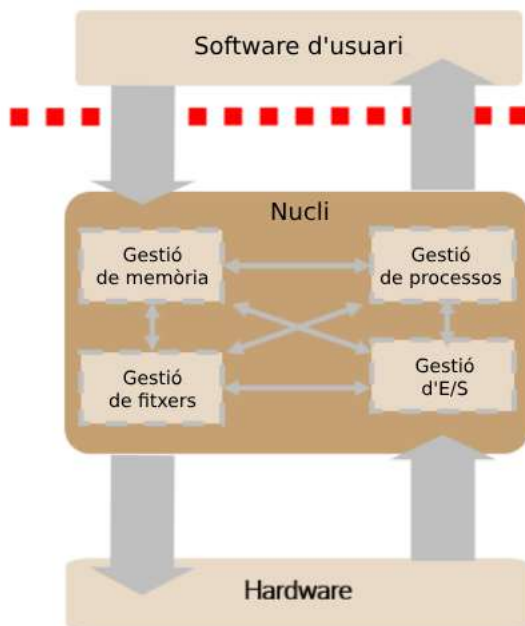


Figura 1.1. Estructura monolítica.

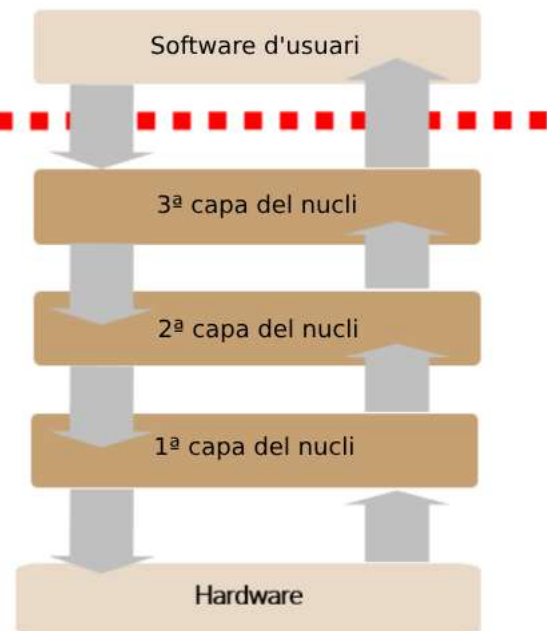


Figura 1.2. Estructura per capes.

El model **microkernel** -o **client/servidor**- organitza les funcions del sistema operatiu en mòduls senzills i aïllats del nucli. Gràcies a això, tenim mòduls per a la gestió d'entrada/sortida, memòria, sistema de fitxers, etc. Aquesta estructura ofereix una major tolerància davant de les fallades, una millor seguretat i més portabilitat entre el maquinari de diferents sistemes.

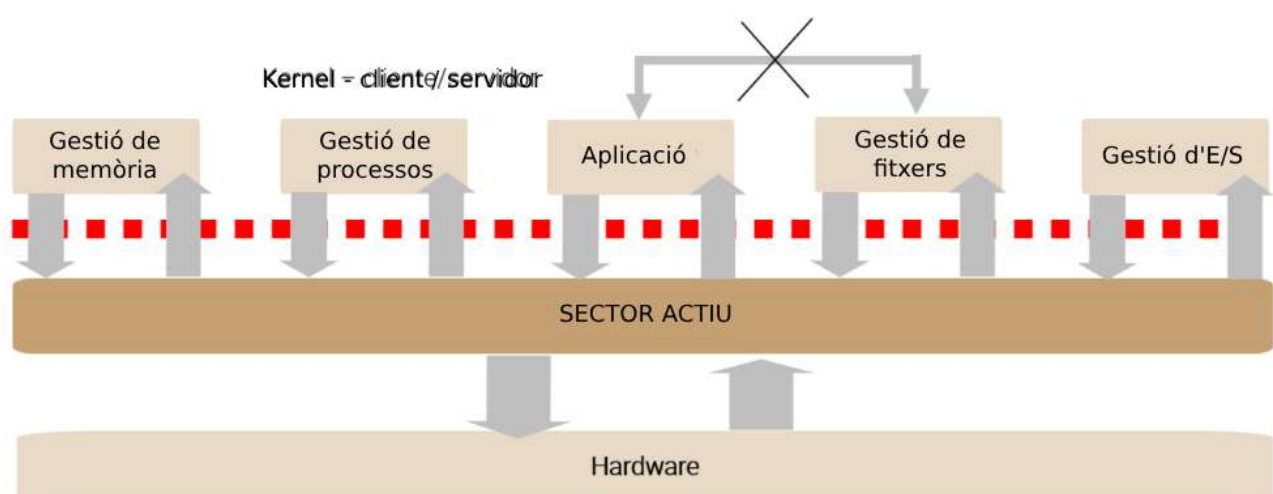


Figura 1.3. Estructura microkernel o client/servidor

Cas pràctic 1

Tenim un bug!

Avui dia fem servir la paraula bug («bestiola», en anglès) per al·ludir a un funcionament anòmal en un programa o en un dispositiu. Però d'on és procedent l'ús d'aquest terme? Investiga'l a Internet.

Solució

Al principi, els ordinadors eren unes màquines enormes, pesades i proveïdes de moltíssims cables i vàlvules de buit.

L'any 1945, els operadors del Harvard Mark II van descobrir que la causa dels errors que estaven obtenint durant el seu treball era una arna atrapada en un relé, la qual cosa provocava errors de contacte i donava lloc, per tant, a l'error.

3. Classificació i tipus.

Els sistemes operatius es poden classificar segons multitud de característiques diferents.

A més, en funció de l'ús que els donem, necessitarem adoptar un tipus de sistema operatiu o un altre.

En aquesta secció veurem diferents característiques, mostrarem quines són les opcions de què disposem i aprendrem a triar el sistema operatiu que necessitem segons els nostres requisits i propòsits.

Important

Tot i que els termes **programa** i **procés** estan relacionats, no són equivalents i convé entendre bé les diferències. Un **programa és un conjunt d'instruccions** que cal seguir per fer una tasca. Els programes s'ubiquen al disc dur. Per la seva banda, un **procés és un programa en execució** que resideix a la memòria principal, és a dir, que tindrà un estat, tasques parcialment realitzades, tasques acabades o fins i tot tasques sense començar. Si utilitzéssim, amb fins il·lustratius, un símil culinari, el programa seria la recepta en què es detallen els passos per seguir i els ingredients necessaris per elaborar un plat, mentre que el procés seria el punt exacte en què ens trobem quan en tenim una recepta a mig fer. Per posar un exemple, això és el que passa quan tenim el sofregit llest, però encara hem de bullir la pasta.

Classificarem els sistemes operatius seguint aquests criteris:

3.1. Segons la utilització dels recursos

Monoprogramats: només admeten un programa carregat al sistema. El programa es carrega a la memòria i s'hi queda fins que acaba l'execució. Durant aquest temps, no pot fer cap altra tasca.

Multiprogramats: són capaços de fer més d'una tasca. Disposen de diferents programes carregats a la memòria. És important diferenciar aquesta noció del concepte de multitasca, en què el sistema operatiu és capaç, a més, de realitzar aquestes tasques simulant que les executa simultàniament (per tenir una multitasca real necessitem disposar de tants processadors com tasques vulguem executar alhora).

- Multitasca apropiativa: és el sistema operatiu el que controla, a cada moment, quin procés fa ús del processador.
- Multitasca cooperativa: és el mateix procés el que decideix si ha de continuar usant el processador o si s'ho cedeix a un altre procés.

3.2. Segons la seva interactivitat

Quan parlem d'interactivitat, ens referim a la manera com es produeix la comunicació entre el sistema operatiu i l'usuari.

- **Sistemes de processament per lots** (batch): són aquells on introduïm un lot d'accions o de passos que el sistema operatiu ha de seguir. El sistema operatiu els processa a l'ordre en què els ha rebut i genera la sortida corresponent. Els primers sistemes operatius que processaven els lots de targetes perforades funcionaven amb aquest tipus de sistemes operatius.
- **Sistemes de temps compartit** (time sharing): aquest tipus de sistemes operatius accepten l'execució de diferents tasques alhora (de manera simulada, ja que, en la majoria dels casos, el nombre de processos en execució és superior al nombre de processadors). Això implica que el processador estarà assignat a un determinat procés durant un temps limitat, anomenat generalment quantum. La major part dels sistemes operatius actuals són sistemes de temps compartit, fins i tot encara que el nostre sistema tingui més d'un processador físic, ja que sempre s'està fent un nombre de processos superior al nombre total de processadors físics disponibles.
- **Sistemes en temps real** (real time): es tracta de sistemes operatius destinats a fer tasques crítiques que depenen del temps de resposta. En ells, el temps màxim de resposta està limitat i acotat; és a dir, abans d'executar una tasca determinada, l'usuari sap quin és el temps màxim que necessitarà el sistema per tornar un resultat. Per exemple, els ordinadors de control de vol d'una aeronau utilitzen aquest tipus de sistemes operatius. I és que no tindria sentit que, en girar la palanca del vehicle, aquest no respongués pel fet que el sistema operatiu es troba fent qualsevol altra tasca menys important i vital. Quan el temps de resposta no és

crític ni és conegut a priori, els anomenem **sistemes en temps diferit**. Aquest tipus de sistemes són els més abundants.

3.3. Segons el nombre d'usuaris

Segons el nombre d'usuaris que poden fer servir el sistema de manera simultània, trobem:

- **Sistemes monousuari**, en què un sol usuari pot fer ús del sistema en un determinat moment. Hi pot haver altres usuaris al sistema, però només un serà capaç d'utilitzar-lo en un moment donat.
- Per contra, els **sistemes multiusuari** permeten, a través de diferents tècniques de multiprogramació, que diferents usuaris accedeixin al sistema (de manera directa o mitjançant connexions remotes) per tal d'executar tasques sobre el sistema.

3.4. Segons el tipus d'aplicació

En funció del tipus d'aplicacions informàtiques que serà capaç d'executar el sistema operatiu, hi podem trobar:

- Sistemes de **propòsit general**: poden executar qualsevol tipus d'aplicació informàtica. Els sistemes operatius més usuals, com ara Windows, GNU/Linux, Mac OS, Android, iOS, etcètera, serien un bon exemple d'aquesta classe.
- Sistemes de **propòsit específic**, dissenyats per executar específicament un tipus d'aplicacions informàtiques o de tasques concretes. Aquests són, per exemple, els sistemes operatius que inclouen les màquines industrials o els electrodomèstics que tenim a les nostres llars.

3.5. Segons el nombre de processadors

Al principi de la informàtica, els sistemes informàtics comptaven amb un sol processador, però avui dia és habitual que molts compten amb diversos d'aquests processadors. En aquest sentit, tenim:

- Sistemes **monoprocessadors**. El sistema compta amb un únic processador i, per tant, aquest és el responsable d'executar tots els processos que es requereixin.

- **Sistemes multiprocessadors o multiprocés.** En aquests, podem trobar més d'un processador, cosa que permet executar més d'una tasca alhora. De la mateixa manera, si una única tasca és divisible, permeten fer un mateix procés en menys temps.

Els nous sistemes operatius es desenvolupen amb metodologies orientades a objectes. Tot al sistema operatiu és una col·lecció d'objectes. El nucli s'encarrega de mantenir les definicions dels objectes i el control relatiu a accedir-hi.

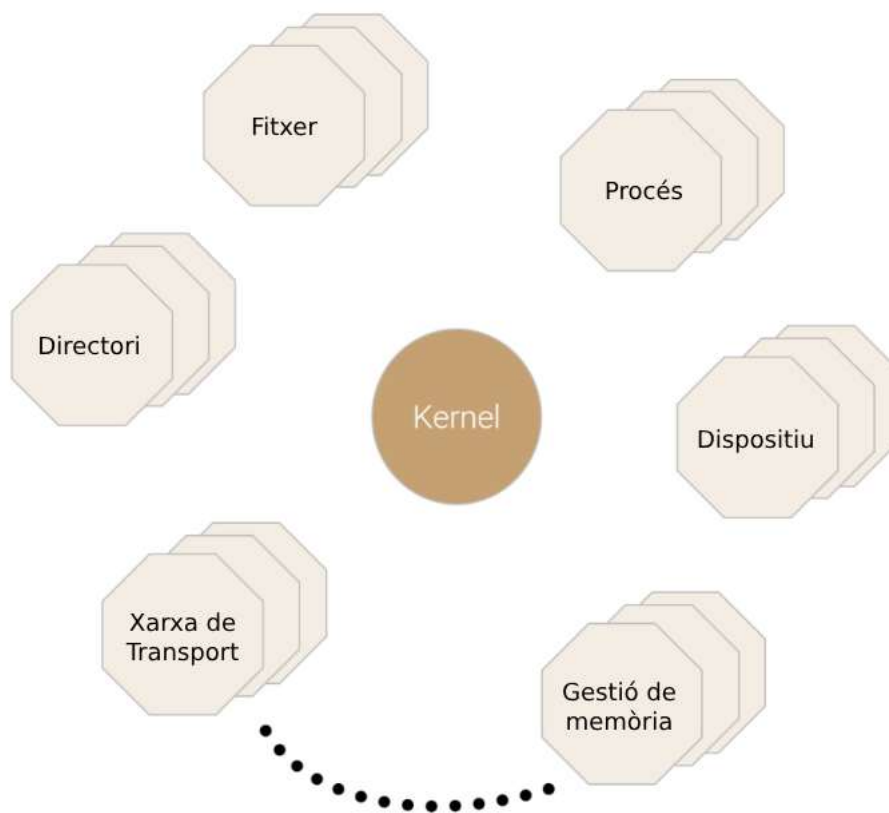


Figura 2.1 Estructura orientada a objectes.

Ara que els sistemes són cada cop més ràpids i que comptem amb infraestructures de xarxa també molt veloços, s'acostuma a combinar diversos processadors en una **estructura multiprocessador** per aconseguir una solució econòmica i escalar d'acord amb les necessitats de cada moment. D'aquesta manera, tenint en compte la relació que hi ha entre la memòria i els processadors, tenim:

- **Multiprocessadors fortament acoblats o sistema de memòria compartida.** En aquest cas, cada processador té accés a la totalitat de la memòria del sistema.

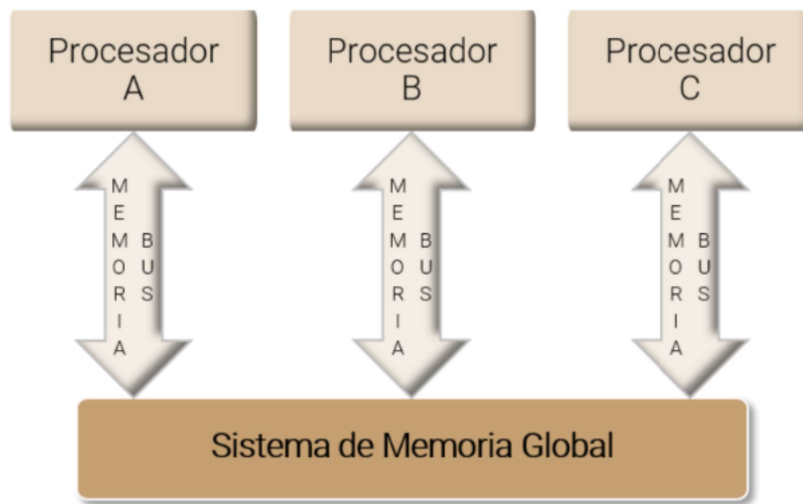


Figura 2.2 Multiprocessadors fortament acoblats.

- **Multiprocessadors feblement acoblats o sistema de memòria distribuïda.** En aquest sistema, cada processador té accés exclusiu a una memòria privada i els processos es comuniquen a través del pas de missatges.

3.6. Segons la distribució de les tasques del sistema

Amb l'avenç de les tecnologies de comunicació, cada cop és més usual que els sistemes operatius cooperin entre si per obtenir millors resultats. Així doncs, podem trobar-hi:

- **Sistemes distribuïts.** Hi ha un conjunt de màquines -amb diferents capacitats i dotades de diversos tipus de processadors i arquitectures- que es comuniquen entre si mitjançant xarxes de comunicacions. D'aquesta manera, es reparteixen el treball d'acord a una subdivisió de les tasques, cosa que permet millorar el rendiment final del sistema en conjunt. A dia d'avui, els grans sistemes d'informació basen la seva estructura en aquest tipus de sistemes, ja que els permeten millorar-ne el rendiment, gaudir d'independència respecte a la ubicació de l'usuari i presentar una millor tolerància davant de les fallades.
- **Sistemes centralitzats,** una sola màquina és la responsable de realitzar totes les tasques que exigeixin al sistema operatiu.

Cas pràctic 2

Aplica els criteris

Per a cadascun dels criteris vistos en aquest apartat, assigneu la tipologia més adequada per als sistemes operatius GNU/Linux i Windows.

Solució

Pel que fa a la utilització de recursos, tots dos són multiprogramats i permeten fixar una multitasca apropiativa.

Pel que fa a la interactivitat, tots dos solen ser de temps compartit, encara que hi ha versions de GNU/Linux de temps real.

Pel que fa als usuaris, tots dos sistemes són multiusuari.

Si parlem del tipus d'aplicacions que podeu executar, tots dos són de propòsit general.

A més, tots dos suporten sistemes multiprocessador.

Finalment, encara que generalment són sistema centralitzats, algunes versions de GNU/Linux es poden fer servir per realitzar tasques de manera distribuïda.

4. Funcions.

El sistema operatiu és el primer programa que entra en funcionament a qualsevol sistema informàtic.

A més, asseu les bases sobre les quals s'executaran la resta d'aplicacions i programes dels usuaris.

En aquest epígraf aprendrem quines són exactament les funcions que realitza el sistema operatiu i coneixerem cadascun dels seus elements.

4.1. Objectius d'un sistema operatiu

Els objectius d'un sistema operatiu són, d'una banda, incrementar la productivitat dels usuaris i facilitar-ne l'ús de l'equip. D'altra banda, tracten així mateix d'oferir un entorn còmode que resulti més abstracte o independent pel que fa al maquinari. També pretén optimitzar la utilització del maquinari per maximitzar-ne el rendiment.

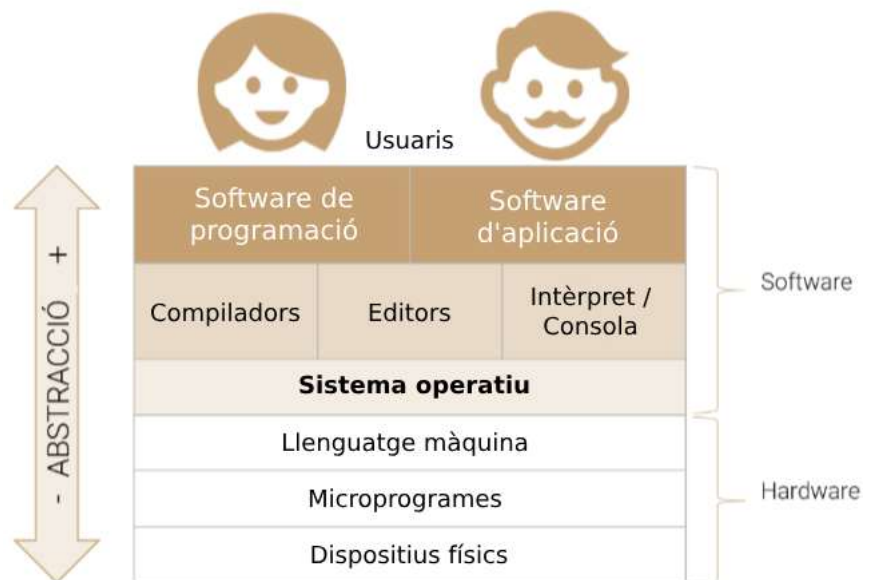


Figura 3.1 Relació entre l'usuari i el maquinari a través del Sistema Operatiu com a intermediari.

Altres propòsits importants que cal esmentar són: Gestionar els recursos del maquinari i del programari decidint qui, quan, com i durant quant de temps s'utilitzarà un recurs per part d'un procés; resoldre conflictes entre peticions concurrents i preservar la integritat del sistema.

4.2. Funcions

Les funcions més importants d'un sistema operatiu són les següents:

- Administrar el processador o els processadors, en cas de comptar amb sistemes multiprocessador.
- Administrar la memòria.
- Comunicar els dispositius amb els usuaris de manera transparent.
- Organitzar les dades, garantint així un accés ràpid i segur.
- Gestionar les comunicacions de xarxa
- Facilitar les entrades i sortides.
- Oferir tècniques de recuperació d'errors.
- Evitar que uns usuaris interfereixin en la feina dels altres.
- Generar estadístiques i fitxers de registre d'esdeveniments del sistema.
- Compartir el maquinari i les dades entre múltiples usuaris.

Totes aquestes funcions es poden resumir en dues funcions primordials i indispensables:

- **Gestionar el maquinari**, cosa que fa referència al fet que és el responsable de gestionar el hardware del sistema de la manera més eficient possible.
- **Facilitar la feina a l'usuari**, ja que permet i facilita a l'usuari la tasca de comunicació amb la màquina

Tot seguit, ens centrarem a detallar algunes d'aquestes funcions de manera més exhaustiva:

A. Gestió de processos

En el punt anterior hem assenyalat la diferència entre un procés i un programa, però ara veurem què fa realment el sistema operatiu amb els processos i quina responsabilitat hi té.

Els processos es poden trobar a diferents estats.

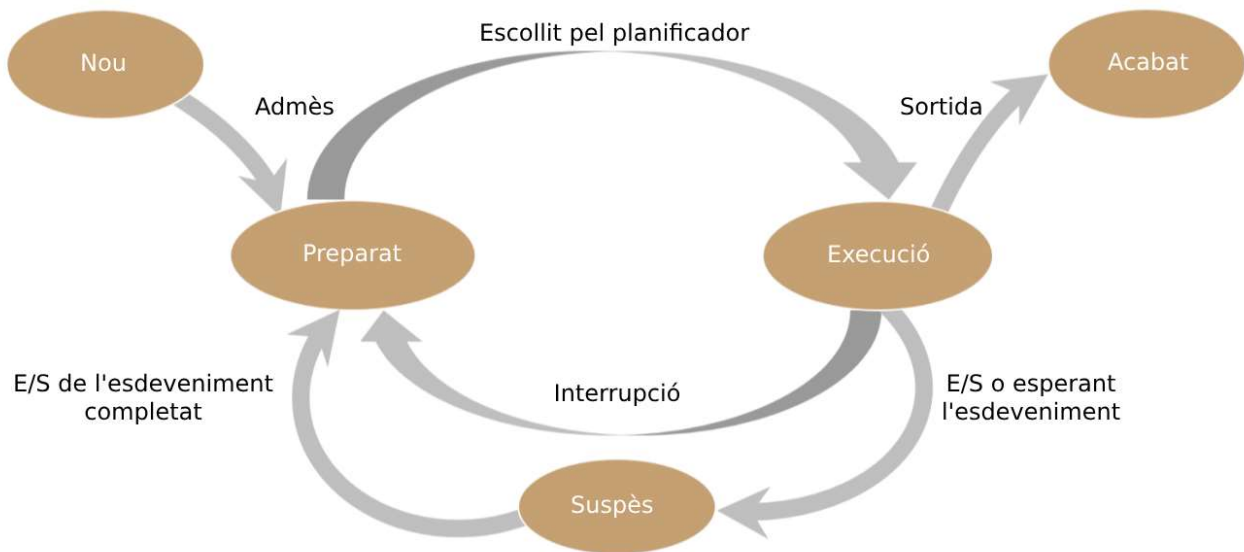


Figura 3.2 Estats de procés.

Quan un usuari decideix iniciar un procés, arriba a l'estat NOU. A continuació, després de ser admès, passa a l'estat PREPARAT. Més endavant, el **planificador** de processos escull el procés que passarà a estat EN EXECUCIÓ.

Quan un procés es troba en execució, poden passar 3 coses:

1. Que el procés acabi de manera normal, passant així a l'estat ACABAT.
2. Que el procés es vegi detingut per una interrupció de sistema, per un procés amb més prioritat o perquè, simplement, s'ha esgotat el seu temps disponible (quantum), de manera que passarà de nou a l'estat PREPARAT en espera que li arribi, de nou, el torn.
3. Que el procés necessiti rebre una entrada o fer una sortida (per exemple, esperar l'entrada de teclat de l'usuari, o la impressió d'un document). En aquest cas, el procés passa a l'estat SUSPÈS, que abandonarà per passar a l'estat PREPARAT tan aviat com l'Entrada, la Sortida o l'esdeveniment que esperava succeeixi.

El **planificador** pot triar el procés següent d'acord amb diferents criteris i algorismes:

- FCFS, o First Come First Served: s'atén primer el procés que arriba abans.
- RR, o Round Robin: el processador s'assigna en torns rotatoris als processos que estan preparats durant un fragment de temps petit anomenat quantum.
- SJF, o Shortest Job First: primer s'atén el procés més curt.
- SRTF, o Shortest Remaining Time First: primer s'executa el procés al qual li quedi menys temps per acabar.

Tot aquest canvi d'estats és responsabilitat del sistema operatiu, que utilitza per dur-los a terme una taula de control de processos o Process Control Block (PCB). La informació que conté el PCB és, entre altres dades, l'identificador del procés i l'identificador del pare (és a dir, de qui l'invoca), l'identificador de l'usuari i del grup al qual pertany l'usuari que llença el procés, la informació sobre l'estat del procés, el planificador, els segments de memòria i els recursos assignats.

Punter	Estat del procés
Número del procés	
Comptador de programa	
Registres	
Límits de memòria	
Llista de fitxers oberts	
...	

Figura 3.3. PCB (Bloc de control de processos).

B. Gestió de memòria

El gestor de la memòria és el responsable de:

- Assignar zones de memòria als processos quan la requereixin i alliberar-la quan hagin acabat.
- Assegurar que un procés no pugui accedir a la memòria que té un altre procés assignat.
- Mantenir un control sobre les zones de memòria assignades i les que no ho estan.

- En els casos en què la memòria principal està totalment ocupada, s'encarrega de fer l'intercanvi (swapping) entre la memòria principal i la secundària.

Es pot fer l'assignació de diferents formes:

- Assignació en **particions fixes**: el sistema operatiu divideix, lògicament, la memòria disponible en particions fixes en el temps (sempre de la mateixa mida, encara que no han de ser totes igual de grans). Un cop fet això, quan arriben els processos, hi ha dues variants:
 - Assignació amb una sola cua per partició. Cada partició hi té una cua de processos que es gestiona de manera independent.
 - Assignació amb una cua comuna a totes les particions. En aquest cas, hi ha una única cua de processos, que van sent assignats a la millor partició disponible a cada moment.
- Assignació en **particions variables** en aquesta variant, el sistema operatiu genera les particions i decideix la mida després de l'arribada del procés, adaptant-ne la mida a les necessitats. Tot i que aquesta situació s'aproxima més a la realitat -ja que el nombre de processos i les seves mides són variables, cosa que permet aprofitar millor la memòria total-, també dóna lloc a un mode de treball més complex per al sistema operatiu.
- Memòria virtual: quan el procés no cap completament (ni encara reservant tota la memòria per a ell), només és possible executar-lo al sistema si fem servir la memòria virtual. El que fa el sistema operatiu és que, a través de diferents tècniques, divideix la memòria que necessita el procés en fragments que va carregant a la memòria principal segons es necessiten. Quan la memòria és plena i es necessita més espai, aquests fragments es descarreguen de la memòria principal i es reserven a la memòria secundària. Les tècniques emprades per fragmentar el procés són les següents:
 - La paginació pura, que consisteix a dividir la memòria principal en un conjunt de particions (totes iguals) anomenades marcs de pàgina. Cada procés es divideix

en fragments de la mateixa mida anomenats pàgines. El sistema operatiu va assignant marcs de pàgina al procés i va carregant-hi les pàgines del mateix, encara que no tenen per què estar junts.

- D'altra banda, la **segmentació pura** es basa que el programador decideix com dividir les parts del seu procés, que es carregaran a la memòria com a segments. A diferència de les pàgines, els segments sí que poden tenir diferents mides.

C. Gestió de l'entrada i sortida (E/S)

La gestió de l'E/S és responsable de facilitar l'intercanvi d'informació entre els perifèrics i l'usuari. Així, podem trobar perifèrics de tres tipus:

- **Entrada:** porten informació des de l'usuari cap al sistema. Ratolí, teclat, webcam.
- **Sortida:** mostren informació del sistema a l'usuari. Per exemple, la pantalla o la impressora.
- **Entrada/Sortida:** realitzen les dues funcions, com passa amb els dispositius d'emmagatzematge o de comunicacions.

Donada aquesta gran varietat de perifèrics, el sistema operatiu utilitza, per simplificar a l'usuari la tasca de comunicació amb aquests perifèrics, els **controladors de dispositius** (device drivers), que actuen com a interfície entre l'usuari i el dispositiu.

Així, podem trobar elements que funcionen com a intermediaris i retenen la informació fins que el dispositiu, molt més lent que el sistema operatiu, és capaç de processar aquesta informació. En aquest sentit, en trobem dos tipus:

- **Spools:** no admeten diferents orígens alhora. Per exemple,

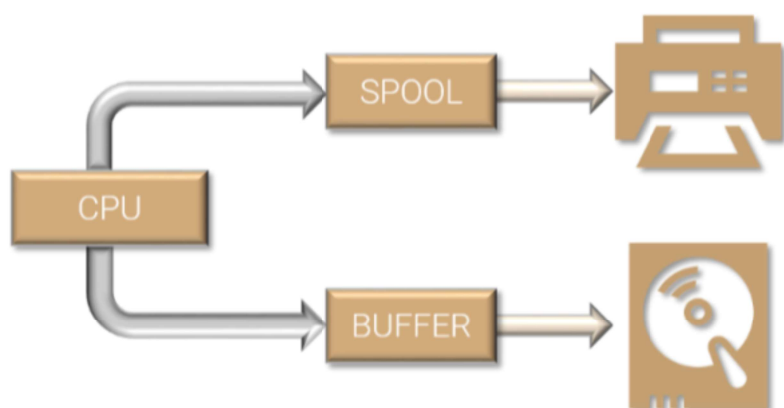


Figura 3.4. Spools i Buffers.

una impressora no pot admetre part d'un treball fins que finalitzi completament l'anterior.

- **Buffers:** sí que admeten diferents orígens simultàniament. Per exemple, les instruccions de gravat o la lectura d'un disc dur.

D. Gestió del sistema de fitxers

Convé diferenciar la gestió que es duu a terme dels arxius i les carpetes a nivell físic -la forma d'emmagatzemar la informació es troba estretament emparentada amb el dispositiu concret emprat a molt baix nivell, cosa que ens porta a fer servir cares, pistes o sectors d'un disc dur- amb el nivell lògic, on el sistema operatiu abstrairà tota aquesta informació perquè l'usuari l'organitzi en fitxers o carpetes.

Els **fitxers** són l'eix central de qualsevol sistema d'emmagatzematge d'informació. Cada fitxer consta d'una sèrie d'atributs que serveixen per identificar-lo i gestionar-lo correctament. Els atributs depenen del sistema de fitxers i, per tant, del sistema operatiu que usem, però generalment en podem trobar.

- Nom: identificat del fitxer.
- Extensió: caràcters que es col·loquen darrere del «.» en alguns sistemes operatius. Indiquen el tipus de fitxer de què es tracta. Per exemple, .txt designa fitxers de text, mentre que mp3 es refereix a fitxers de so.
- Permisos: indica qui pot llegir, modificar, executar... el fitxer.
- Creador, qui va crear el fitxer originalment.
- Data de creació: data i hora de creació del fitxer.
- Data d'última modificació: data i hora en què es va alterar el fitxer per darrera vegada.
- Mida actual: mida, en bytes, que ocupa el fitxer.

D'altra banda, tenim les **carpetes**, uns fitxers especials que serveixen per organitzar la informació, ja que poden contenir al seu interior fitxers o altres carpetes. D'aquesta manera, s'aconsegueix una estructura jeràrquica en què tots els fitxers i totes les carpetes

depenen d'un nexa comú anomenat arrel, de la qual depèn tot el sistema de fitxers. Les carpetes també tenen els seus propis atributs (coincideixen, en gran part, amb els atributs dels fitxers).

E. Carpetes especials

Dins del sistema de fitxers de la majoria de sistemes operatius (inclosos els de Microsoft i GNU/Linux), hi ha dues carpetes especials, que són:

- . (un punt): fa referència a la carpeta on ens trobem actualment.
- ..(punt, punt): fa referència a la carpeta pare de la carpeta actual, de la qual depèn.

Cas pràctic 3

Responsabilitats

Un cop estudiant aquest apartat, sabries dir quin element del sistema informàtic és el responsable últim que un arxiu s'emmagatzemi correctament al disc dur o que s'escolti correctament una cançó pels altaveus?

Solució

En tots dos casos, el **sistema operatiu** ofereix una interfície a l'usuari perquè, de manera molt senzilla, pugui realitzar les dues tasques esmentades, amb independència de com de complicades puguin resultar a baix nivell.

D'aquesta manera, l'usuari no ha de plantejar-se a quina cara, pista o sector vol emmagatzemar un fitxer. Així mateix, tampoc no està obligat a pensar en la conversió — des del format digital al format analògic— que s'ha de realitzar per poder escoltar un fitxer mp3.