

Introducció als sistemes operatius

José Ramón Herrero Zaragoza
Enric Morancho Llena
Dolors Royo Vallés

PID_00169367

Índex

Introducció.....	5
Objectius.....	6
1. Classificació del programari. Definició de sistema operatiu..	7
2. Visió històrica dels sistemes operatius.....	10
2.1. Els primers sistemes computadors (la primera generació)	10
2.2. La segona generació	11
2.3. La tercera generació	13
2.4. De la quarta generació fins ara	16
3. Els serveis que ofereix el sistema operatiu.....	19
3.1. Les crides al sistema	20
3.2. L'interpret d'ordres	20
Resum.....	22
Activitats.....	23
Glossari.....	24
Bibliografia.....	25

Introducció

Un sistema computador és format pels elements següents:

- El maquinari (*hardware*), que fa referència a la màquina física.
- El programari (*software*), que fa referència al conjunt de programes que s'executen i que permeten treure profit de les prestacions que ofereix el maquinari.

Sense programari el computador és pràcticament un objecte sense cap utilitat. El programari permet emmagatzemar informació, processar-la i recuperar-la. En general, també permet fer una sèrie d'activitats molt diverses que justifiquen la inversió econòmica en el sistema computador.

Actualment, un sistema computador consta d'un o més processadors, memòria, terminals, discos, impressores, targetes de xarxa, etc. Els dispositius que conté són de tipus diversos (magnètic, mecànic, làser, sense fils, etc.), tenen un funcionament molt variat i en poc temps poden esdevenir obsolets, per la qual cosa seran substituïts per nous tipus de dispositius. A més a més, en general hi ha més d'un usuari que utilitza alhora el sistema computador. Així, si un usuari vol fer servir el sistema computador de manera eficient i amb una certa protecció de la seva informació respecte de la resta d'usuaris, difícilment ho podrà aconseguir sense cap tipus d'ajuda. És evident que hi ha d'haver alguna manera d'aconseguir que els usuaris puguin utilitzar el sistema computador sense que hagin de ser conscients de la complexitat del maquinari.

El problema s'ha resolt posant per sobre del maquinari una capa de programari amb l'objectiu de gestionar les diferents parts del sistema computador de manera eficient i, alhora, presentar a l'usuari una màquina virtual molt més senzilla de comprendre i utilitzar. Aquesta capa de programari és l'anomenat *programari de sistema*, la part més important del qual és el sistema operatiu (SO).

SO

Utilitzem les sigles SO com a abreviació de *sistema operatiu*.

Objectius

En els materials didàctics d'aquest mòdul presentem els continguts i les eines necessàries per a assolir els objectius següents:

- 1.** Tenir una visió com més clara millor del que és un sistema operatiu. A causa de la complexitat del concepte, formulem la definició a partir de la descripció de les funcions que aconsegueix el sistema operatiu dintre del sistema computador, i a partir del procés evolutiu que han experimentat els sistemes operatius al llarg de la història, des dels primers computadors fins als nostres dies.
- 2.** Comprendre la visió externa del sistema operatiu: des del punt de vista de l'usuari que utilitza el sistema operatiu per a consultar, modificar, crear, destruir i emmagatzemar informació, i des del punt de vista de l'usuari com a programador d'aplicacions que s'executaran sobre aquest sistema operatiu.

1. Classificació del programari. Definició de sistema operatiu

En la figura 1 es mostra un esquema general del programari que inclou un sistema computador:

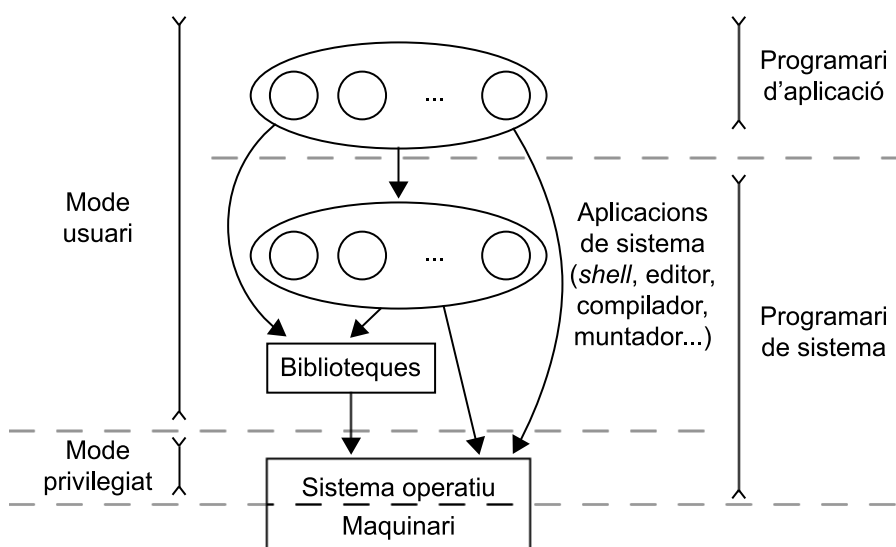


Figura 1. Esquema d'un sistema computador en què es mostra una classificació dels tipus de programaris

La capa inferior correspon al **maquinari** del computador. Per sobre d'aquesta trobem el **sistema operatiu**. És molt difícil donar una definició exacta i acurada del terme *sistema operatiu*, el qual generalment es defineix a partir de les funcions que desenvolupa. Bàsicament són les dues funcions que presentem tot seguit:

1) Gestió eficient dels recursos del sistema:

L'SO controla l'accés eficient als recursos del computador: la memòria principal, la unitat central de procés (CPU, processador) i els dispositius. És a dir, el SO s'encarrega principalment de tasques de protecció i d'utilització eficient del sistema computador.

En general, en un sistema computador tenim molts programes que s'executen al mateix temps, són programes que poden pertànyer a un o a diversos usuaris, els quals competeixen pels diferents recursos del sistema (disc, memòria, etc.). El sistema operatiu reparteix el temps de CPU entre els diferents programes i aconsegueix una execució concurrent, protegeix l'accés a la memòria¹ i coordina l'accés als dispositius (discos, memòria, impressora, etc.).

⁽¹⁾El SO protegeix l'accés a memòria entre diferents usuaris i entre usuaris i el mateix SO. Això s'estudiarà en detall en el mòdul 3.

2) Presentació als usuaris d'una màquina virtual molt més senzilla d'utilitzar:

El sistema operatiu proporciona un entorn de treball a l'usuari i als programes d'aplicació que permet emprar el computador (executar programes) de manera més fàcil i intuïtiva. Des d'aquest punt de vista, el sistema operatiu proporciona a l'usuari una màquina virtual molt més fàcil d'entendre i utilitzar pel fet que amaga la complexitat del maquinari.

Lectura d'un fitxer

Per exemple, si no existís el sistema operatiu, a l'hora de llegir un fitxer l'usuari necessitaria comprendre i ocupar-se dels detalls tècnics del disc: com ha estat formatat (FAT, NTFS, EXT2...), geometria del disc (nombre de cares, pistes, sectors, etc.), posicionament dels capçals, lectura, etc. El sistema operatiu ens permet fer la lectura de dades d'un fitxer amb una operació molt més senzilla i intuïtiva, que no requereix a l'usuari cap coneixement detallat del funcionament del disc. Amb l'operació `read(nom_fitxer, pos, reg)` s'indica que es vol llegir del fitxer `nom_fitxer` el registre `reg`, que és a la posició `pos`. Més endavant veurem que aquestes operacions reben el nom de *crides al sistema* (*system calls*), i el conjunt de crides al sistema ofertes pel sistema operatiu constitueixen l'API (*application programming interface*) del sistema operatiu.

Per sobre del sistema operatiu tenim la **resta de programari de sistema**, el qual consta de biblioteques de rutines i d'aplicacions de sistema com ara l'interpret d'ordres (*shell*), compiladors, editors i, en general, programes que faciliten la comunicació entre el sistema operatiu i l'usuari.

Un usuari pot tenir la sensació que l'interpret d'ordres i les altres aplicacions de sistema també formen part del sistema operatiu perquè aquestes aplicacions s'instal·len conjuntament amb el sistema operatiu. A més, molts usuaris no han de ser conscients de l'existència de les crides al sistema perquè interaccionen amb el sistema operatiu de manera indirecta utilitzant aquestes aplicacions. És important adonar-se de la diferència entre el sistema operatiu i la resta de programari de sistema:

1) Una manera d'adonar-se de la frontera existent dins del programari de sistema és que un usuari qualsevol d'un sistema computador no hauria d'estar obligat a utilitzar les aplicacions de sistema disponibles; si vol, n'hauria de poder utilitzar d'altres o fins i tot implementar-les ell mateix, sense necessitar cap autorització especial. Per exemple, l'usuari pot seleccionar l'interpret d'ordres, l'editor, el compilador, etc. que vol utilitzar. En canvi, tots els usuaris estan obligats a utilitzar el programari que forma part del sistema operatiu; únicament usuaris autoritzats poden introduir modificacions dins del codi del sistema operatiu.

Biblioteques de rutines

En alguns textos escrits en català o en castellà es fa servir el terme *llibreria* en comptes del terme *biblioteca* a causa d'una traducció incorrecta del terme anglès original (*library*). Una biblioteca de rutines és un conjunt de rutines d'ús comú i que es troba disponible per a tots els usuaris.

2) Una altra diferència entre el sistema operatiu i la resta de programari és que, al cap i a la fi, tot el programari s'executa sobre el processador com a instruccions de llenguatge màquina. Per a garantir que els sistemes operatius puguin fer la feina correctament, cal que el processador sigui conscient de quan està executant codi propi del sistema operatiu. Això és degut al fet que el codi propi del sistema operatiu hauria de poder dur a terme operacions "privilegiades", mentre que la resta del programari no. Els processadors disposen de diversos modes d'execució: un mode d'execució "privilegiat" per a executar codi propi del sistema operatiu, i un mode "no privilegiat" (usuari) per a executar qualsevol altre codi (biblioteques, aplicacions de sistema, aplicacions d'usuari).

3) Tot i això, és possible que en algun document trobeu que es parli de *sistema operatiu* per a referir-se a tot el programari de sistema presentat en la figura 1. En aquest cas, s'acostuma a utilitzar el terme *nucli* o *kernel* del sistema operatiu per a referir-se a la capa que es troba immediatament a sobre del maquinari del sistema computador.

Finalment, a sobre del programari de sistema hi ha el **programari d'aplicació**. Una part d'aquest programari la fa servir l'usuari sobretot per a resoldre problemes específics. En aquest nivell trobem fulls de càlcul, processadors de textos, navegadors web, lectors de correu electrònic, jocs, etc.

Programari i llenguatge màquina

Trobareu més informació sobre el programari i la seva execució sobre el processador com a instruccions de llenguatge màquina al mòdul 2.

2. Visió històrica dels sistemes operatius

Per a entendre bé què és un sistema operatiu i per a què serveix ens pot ajudar conèixer-ne l'evolució. A continuació expliquem com han evolucionat els sistemes operatius des de les versions més primitives (monitors residents i sistemes operatius per lots), passant pels sistemes operatius multiprogramats i amb temps compartit, fins als sistemes operatius distribuïts i els sistemes operatius en xarxa, els sistemes operatius en temps reals i per a dispositius embastats.

No és fàcil separar el desenvolupament de l'arquitectura dels computadors de l'evolució dels sistemes operatius, essencialment perquè els sistemes operatius han evolucionat en funció dels canvis tecnològics intentant treure el màxim profit de les noves tecnologies. Al seu torn, aquestes han donat solucions a molts problemes que han anat sorgint en els sistemes operatius.

2.1. Els primers sistemes computadors (la primera generació)

Al principi només hi havia el maquinari del computador. La tecnologia utilitzada en la construcció dels primers computadors eren les vàlvules de buit, que obligaven a fer uns aparells monstruosament grans.

En aquesta primera època no hi havia sistema operatiu i l'usuari interaccionava directament amb el maquinari del computador. Per a programar aquests aparells calia configurar manualment un seguit d'interruptors i de quadres de connexió. Evidentment, la programació de la màquina a aquest nivell era molt poc productiva, tant per a l'usuari com per al maquinari. El procés d'introducció de programes, que era llarg i complicat, feia inviable l'execució de programes mitjans i grans en aquest entorn.

Aquests primers computadors s'utilitzen per a efectuar càlculs matemàtics en qüestions militars. Per exemple, per a calcular trajectòries balístiques i taules de sinus i cosinus.

Cap al final d'aquesta època comencen a aparèixer dispositius com les impresores de cintes de paper i els lectors de targetes perforades. Això va permetre codificar els programes utilitzant targetes perforades i emmagatzemar el resultat en cintes de paper o bé en targetes perforades.

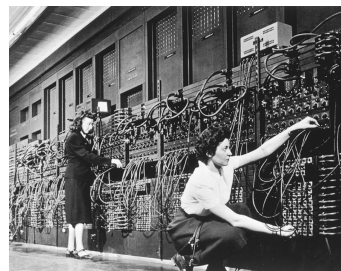


Figura 2

L'ENIAC, *Electronic numerical integrator and computer* (computador i integrador numèric electrònic), pesava aproximadament trenta tones, ocupava 150 m² i consumia una potència de 150 kW.

2.2. La segona generació

L'aparició del transistor en la dècada de 1950 va permetre construir computadors molt més petits i més fiables, de manera que ja es podien fabricar amb la idea de vendre'ls. Eren màquines molt cares i només eren a l'abast de les grans corporacions, institucions com l'exèrcit, les universitats i el govern.

Paral·lelament, constatem els fets següents:

- Apareixen dispositius d'entrada/sortida nous: els lectors de targetes perforades i les impressores.
- També s'inventen dispositius d'emmagatzematge nous: les unitats de disc i les cintes magnètiques.
- Es desenvolupa el primer programari de sistema, que inclou assembleadors, compiladors de llenguatges de programació d'alt nivell², carregadors, muntadors (*linkers*), biblioteques de funcions matemàtiques i rutines estàndard d'entrada/sortida dels diferents dispositius que poden utilitzar els programes³.

⁽²⁾Algol, Cobol, Fortran.

⁽³⁾Impressores, monitors i targetes perforades.

Amb aquest nou entorn es millora considerablement el procés de programació i execució de programes respecte de l'etapa inicial. En aquesta etapa el procés té lloc seguint els passos següents:

1) Els programes s'escriuen en llenguatges assembleadors o d'alt nivell; un cop creats s'anomenen *codi font* (*source code*).

2) Els programes es tradueixen de manera automàtica mitjançant els compiladors/traductors (aplicacions de sistema) a llenguatge màquina:

- Si hi ha errors de sintaxi, cal corregir-los i tornar a fer la traducció.
- Si no hi ha errors sintàctics, el codi objecte produït a partir del codi font pel traductor/compilador es pot carregar a memòria i executar.

3) Una altra aplicació de sistema, el carregador, automatitza el procés de carregar el programa a memòria per a executar-lo.

4) Després de transferir el control al programa carregat per mitjans manuals o automàtics, comença l'execució del programa.



Figura 3

Les targetes i les cintes perforades es feien servir per a programar els computadors i per a rebre els resultats dels seus càlculs mitjançant màquines lectores-perforadores de targetes.

Un cop carregat a memòria, el programa es pot reexecutar amb diferents dades, i els resultats es treuen per la impressora o en cinta de paper. Si es detecten errors en temps d'execució és possible determinar on s'han produït fent un abocament de la memòria i dels registres del computador o bé utilitzant el programa depurador (*debugger*).

En aquest sistema, les rutines d'entrada/sortida, juntament amb el programa carregador, es poden considerar una primera versió molt rudimentària del sistema operatiu. Els traductors de llenguatge d'alt nivell, els editors i els depuradors són aplicacions de sistema que s'inclouen en els diferents serveis que ofereix el sistema operatiu, però generalment es considera que no en són part.

Malgrat totes aquestes millores, el mode d'operació continuava essent poc eficaç a causa de les diverses operacions manuals que calia efectuar, les quals comportaven una utilització baixa dels recursos del sistema. Com que el cost econòmic del sistema computador era elevat, no és estrany que es busquessin maneres d'aprofitar el temps perdut i augmentar la utilització dels recursos del sistema a fi d'eliminar els temps morts generats per les operacions manuals.

El primer objectiu va ser automatitzar al màxim totes les operacions utilitzant maquinari. Però, malgrat l'automatització d'aquestes operacions i la millora dels dispositius⁴, la velocitat de les operacions d'entrada/sortida de dades continuava essent massa lenta en comparació amb la velocitat del processador i la de la memòria.

⁽⁴⁾Utilització de cintes magnètiques en lloc de targetes perforades

Com que no era possible reduir el cost temporal de les operacions d'entrada/sortida, es va intentar almenys efectuar-les el mínim nombre de cops possible. La idea era agrupar els treballs de manera que les operacions auxiliars tan sols es fessin un cop per a tots els treballs del grup. Aquest sistema s'anomena **sistema per lots** (sistema *batch*).

El sistema per lots podria funcionar, per exemple, d'aquesta manera: agrupant en un lot tots els programes escrits en llenguatge d'alt nivell Fortran només s'ha de carregar un sol cop el compilador de Fortran, i els treballs que es fan dintre del lot es poden processar l'un darrere l'altre amb un seqüenciamen automàtic. Perquè sigui possible aquest seqüenciamen automàtic, s'ha de poder indicar al sistema operatiu almenys quan comença i quan finalitza un programa dintre del lot. En aquest punt cal introduir el concepte de llenguatge de control de treballs (JCL⁵). Amb aquest llenguatge es podia indicar al sistema operatiu l'inici i la finalització d'un treball, donar ordres de càrrega i execució i sol·licitar certs recursos del computador (temps de CPU, quantitat de memòria, etc.).

⁽⁵⁾Les sigles JCL provenen de l'abreviació de l'expressió anglesa *job control language*.

En els computadores de la segona generació, l'execució dels programes era totalment seqüencial. Quan un programa es parava per a esperar la finalització d'una operació d'entrada/sortida (d'una unitat de cinta o de qualsevol altre dispositiu), la unitat central de procés (CPU) senzillament s'aturava i també s'esperava que l'operació d'entrada/sortida acabés. En el cas de programes científics, en què domina el càlcul d'operacions en coma flotant (el qual s'efectua en la CPU) i les operacions d'entrada/sortida són poc freqüents, aquest sistema de treball ja era prou eficient.

Malgrat aquesta automatització, la CPU continuava estant molt temps parada. El problema principal era la diferència de velocitats entre la CPU i els dispositius: les CPU en aquest període eren lentes, però els lectors de targetes encara ho eren més. Per tant, la CPU estava molt temps aturada esperant que finalitzés una operació d'entrada/sortida. Per a intentar reduir al màxim aquestes diferents velocitats de treball, va aparèixer el **treball fora de línia** (*off-line*). Amb aquest sistema s'intentava encavalcar les operacions d'entrada/sortida amb l'execució de programes. Mentre els programes escrits en targetes es llegien i es passaven a cinta en una màquina, en una altra màquina es podien executar altres programes que ja havien estat carregats anteriorment a una altra unitat de cinta.

Posteriorment es van idear altres tècniques d'encavalcament com l'**emmagatzematge a la memòria intermèdia** (*buffering*), que consisteix a encavalcar les operacions d'entrada/sortida d'un programa amb les operacions de càlcul del mateix programa.

L'aparició del disc, dispositiu d'accés directe, proporcionà una nova tècnica de treball: la **gestió de cues** (*spooling*), la qual permetia encavalcar les operacions d'entrada/sortida de diferents treballs mentre s'estaven executant altres operacions.

2.3. La tercera generació

La tercera generació de computadors sorgeix a mitjan anys seixanta i tecnològicament es basa en els circuits integrats. Aquestes màquines noves són molt més petites que les de la segona generació, i també són molt més ràpides.

Amb l'aparició de computadors més petits i fiables gràcies a les noves tecnologies i a la possibilitat d'incrementar la producció de computadors, els costos de fabricació s'abarateixen, i així neix un sector informàtic comercial molt important. Els computadors es posen a l'abast d'empreses mitjanament grans i comença l'ús comercial de la informàtica.

Un altre aspecte molt important d'aquesta nova generació és el fet que s'introdueixen mecanismes nous: interrupcions, protecció, cerca d'una concurrència màxima d'operació, etc. Paral·lelament es desenvolupen el maquinari i el programari, que van donar lloc als elements següents:

- Perifèrics més efectius i ràpids.
- Terminals remots (amb els quals es pot accedir a bases de dades, actualitzar comptes corrents, etc.).
- Llenguatges de programació d'ús general, pensats perquè poguessin ser utilitzats per diferents computadors amb característiques de maquinari molt diverses. S'estandarditzen els llenguatges d'alt nivell ja existents (Fortran, Algol, Cobol, PL/1) i n'apareixen d'altres, com el Basic i el Pascal.



Figura 4

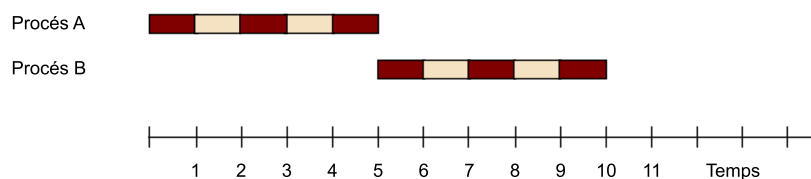
La sèrie IBM 360. El primer grup de computadors construïts amb circuits integrats va ser el de la sèrie 360 d'IBM.

Una de les tècniques més innovadores d'aquesta generació és la **multiprogramació**, que permet encavalcar l'execució de diversos programes.

En el processament de dades comercials el temps d'espera per raó d'operacions d'entrada/sortida pot representar el 90% del temps total d'execució. Si el nostre computador està executant únicament un programa, estarem desaprofitant el processador tot aquest temps d'espera. La solució del problema serà la introducció de la multiprogramació, és a dir, la capacitat de multiplexar la utilització del processador entre diversos programes de manera que, aparentment, el processador estigui executant diversos programes simultàniament.

Durant l'execució d'un programa, generalment podem distingir entre fases de càlcul intensiu i fases en què s'efectuen operacions d'entrada/sortida de manera intensiva. Això s'il·lustra en la figura 5, en què les fases de càlcul intensiu s'indiquen mitjançant rectangles grisos i les d'entrada/sortida amb rectangles blancs. L'execució seqüencial de dos programes es mostra en el punt *a*. *Execució seqüencial*. Per a simplificar-ho, suposarem que els programes tenen un comportament idèntic.

a. Execució seqüencial



b. Execució concurrent

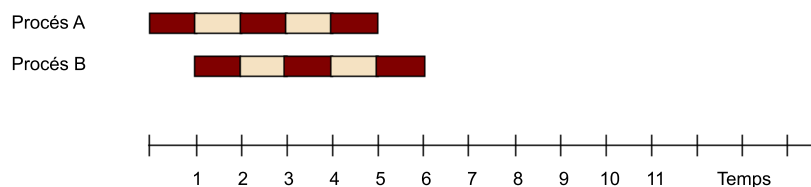


Figura 5. Rectangles foscos: fases de càlcul intensiu. Rectangles clars: fases d'entrada/sortida.

Tal com es pot deduir de la figura 5, durant l'execució seqüencial de tots dos programes la CPU o bé els dispositius d'entrada/sortida estan inactius en algun moment. Una manera de solucionar aquest problema és assignar alguna altra tasca a la CPU o als dispositius d'entrada/sortida quan, per algun motiu, estiguin inactius. Si és possible executar els programes de manera concurrent, aquests poden proporcionar feina tant a la CPU com als dispositius en qualsevol moment durant l'execució dels dos programes.

En el punt *b*. *Execució concurrent* de la figura 5 es mostra un possible escenari de l'execució concurrent dels dos programes. El processador comença executant el primer procés, el *procés A*, i, quan aquest porta a terme una operació d'entrada/sortida, el processador, en lloc de quedar-se inactiu, passa a executar el segon procés, el *procés B*, que se suposa que es troba en la memòria esperant

per a ser executat. Quan aquest segon procés s'acaba o bé sol·licita una operació d'entrada/sortida, el processador torna a executar el *procés A*. El procés continua alternant l'execució dels dos processos fins que finalitzen tots dos.

Com es pot veure en l'esquema de la figura 5, amb l'execució concurrent, o multiprogramada, es pot incrementar considerablement el rendiment del computador: el temps d'execució dels dos programes queda reduït de deu unitats en l'execució seqüencial a sis unitats en l'execució concurrent, i la utilització del sistema en aquest segon cas és d'un cent per cent. L'exemple que hem presentat, encara que és molt representatiu, no és real, ja que generalment la distribució de les fases de càlcul i programació són més variables, de manera que no sempre s'aconsegueix que el processador i els dispositius estiguin actius.

Per a augmentar la utilització dels recursos del sistema computador els sistemes multiprogramats permeten l'execució concurrent de més de dos programes, dels quals es diu que estan competint pels recursos del sistema. El nombre de programes que estan activament en competència pels recursos del sistema determina el **grau de multiprogramació**.

L'aparició de la multiprogramació permet desenvolupar sistemes amb múltiples usuaris connectats simultàniament mitjançant terminals a un sistema computador. En aquest cas, el sistema multiprogramat pur no és gaire eficient, ja que, tal com hem explicat anteriorment, la CPU deixa d'executar un programa i passa a executar-ne un altre que estigui pendent d'execució sempre que el programa que s'està executant fa una operació d'entrada/sortida o bé quan finalitza. A causa d'aquest funcionament, el temps de resposta que té un usuari que està connectat interactivament pot ser considerablement llarg.

Per a solucionar aquest problema es proposa un nou mecanisme anomenat **temps compartit** (*time sharing*). Amb aquest mecanisme els diferents programes que competeixen pels recursos del sistema, entre els quals hi ha la CPU, tenen assignada la CPU durant un període de temps limitat, anomenat *quàntum*⁶. Així, quan un programa deixa d'executar-se, allibera la CPU per algun dels tres motius següents:

- perquè el programa finalitza,
- perquè el programa fa una operació d'entrada/sortida,
- perquè ha passat un quàntum de temps en execució.

El quàntum de temps és prou petit perquè els programes dels diferents usuaris que estan connectats al sistema progressin concurrentment i cada usuari tingui la impressió que és l'únic que utilitza el sistema.

En aquesta generació podem emmarcar el naixement del sistema operatiu UNIX, precursor del sistema operatiu Linux. Ken Thompson, un científic dels laboratoris Bell, va decidir implementar una versió simplificada d'un ambició –però sense èxit– projecte de sistema operatiu anomenat MULTICS. Les bones

⁽⁶⁾En el sistema operatiu Linux, la durada del quàntum és un paràmetre de configuració que es pot triar entre els valors 1 ms, 4 ms i 10 ms.

crítiques de la primera versió de l'UNIX el van encoratjar a escriure una nova versió per a una de les màquines més populars de la època (PDP-11) i, per primer cop en la història dels sistemes operatius, utilitzant un llenguatge d'alt nivell (el llenguatge C, creat per a l'ocasió). Com que l'AT&T (propietària dels laboratoris Bell) no estava autoritzada a fer negoci en el sector informàtic, cap al 1974 va començar a distribuir el codi font de l'UNIX a les universitats que ho demanaven. A partir d'aquest moment, moltes universitats varen començar a utilitzar, estudiar i millorar l'UNIX. Cap al 1984, AT&T va rebre l'autorització per a operar en el sector informàtic, i va començar a comercialitzar l'UNIX System III i, posteriorment, l'UNIX System V. Ara bé, la Universitat de Berkeley també va distribuir la seva versió de l'UNIX (BSD), que entre d'altres millores incorporava el protocol de comunicacions TCP/IP, que ha resultat clau en el desenvolupament de la xarxa Internet. A més, moltes empreses varen comercialitzar les seves versions de l'UNIX (HP-UX, AIX, Solaris, Sun-OS, Ultrix, etc.). Aquesta gran família de sistemes operatius no era totalment compatible, per la qual cosa varen aparèixer diversos esforços per a estandarditzar el sistema operatiu UNIX. L'esforç més destacat va ser POSIX (*portable operating system interface*).

2.4. De la quarta generació fins ara

Amb l'aparició de la integració a gran escala (LSI/VLSI) s'inicia l'era dels computadors personals (1974), als quals s'apliquen tots els mecanismes desenvolupats fins ara (tant de maquinari com de programari). Segurament l'aportació més important d'aquesta nova generació són les xarxes de computadors (Internet i Intranet).

A partir d'aquest moment i fins als nostres dies, els computadors s'utilitzen en qualsevol àmbit: els usuaris a casa seva (computador personal) o al carrer (PDA, telèfons mòbils, etc.), electrodomèstics, les petites i grans empreses (programari de gestió), les corporacions, les universitats i els centres d'investigació (supercomputadors i multicomputadors). En general, hi ha una explosió de recursos: nous dispositius més i més avançats (impressores làser, comunicacions sense fil, etc.) i CPU cada cop més ràpides, sistemes amb més memòria i discos amb més capacitat.

Sorgeixen els **sistemes operatius en xarxa i els distribuïts**, que permeten emprar recursos de màquines remotes (discos, CPU, programari, etc.) amb diferents nivells d'abstracció.

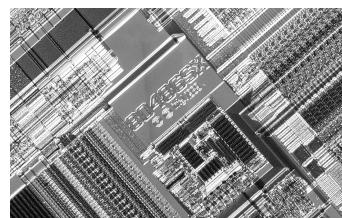


Figura 6
Interior del processador 80486 d'Intel

També s'introdueix el concepte de **sistema operatiu en temps real** referit als sistemes que han de processar dades, que generalment s'obtenen de l'exterior (sensors), en un temps limitat i dintre de períodes de temps determinats. És important veure que aquests sistemes operatius se centren a aconseguir un temps de resposta efectiu, mentre que la utilització adequada de recursos (gestió eficient de la memòria) i la perspectiva de l'usuari queden en un segon pla.

Aplicacions

Els sistemes operatius en temps real s'utilitzen en entorns industrials, en equips de commutació telefònica, en control de vol en avions i en simulacions en temps real.

En aquesta generació podem situar l'aparició dels **sistemes operatius encastats** (*embedded*). Aquests sistemes operatius s'utilitzen en computadors que controlen aparells com ara televisors, reproductors d'MP3, reproductors de DVD, rentadores, telèfons mòbils senzills, etc. Aquests dispositius estan dissenyats per a un ús molt concret i l'usuari difícilment hi podrà instal·lar nou programari. El fet que aquests sistemes no siguin de propòsit general permet simplificar el disseny dels seus sistemes operatius.

També podem situar l'aparició dels **sistemes operatius per a "computadors de butxaca"** (*handheld computers*). Aquí podem incloure un seguit de sistemes computadors pràcticament de propòsit general: PDA, telèfons mòbils, consols, etc. Aquests sistemes operatius cada cop són més sofisticats perquè cada cop incorporen més prestacions, permeten que l'usuari instal·li nou programari i han de controlar perifèrics com ara pantalles gràfiques, pantalles tàctils, càmeres de fotos, GPS, comunicacions sense fil, etc. A més, a diferència d'altres entorns, han de gestionar l'energia de manera eficient per a augmentar el temps d'autonomia dels equips.

Cal assenyalar que segons quina utilització es faci del sistema computador interessa que el sistema operatiu proporcioni eines específiques als usuaris que l'han d'utilitzar. Així, doncs, s'han d'aconseguir els objectius següents:

- Els sistemes en xarxa o els distribuïts han de desenvolupar gestors de comunicació eficients.
- Els sistemes multiprogramats/multiusuari s'han de centrar a aconseguir temps de resposta acceptables.
- Els sistemes en temps real interessa que siguin com més senzills millor per a poder processar dades de l'exterior tan ràpid com sigui possible.

En aquesta generació podem emmarcar el naixement de dos dels sistemes operatius més utilitzats en l'àmbit domèstic: la versió 1.0 del sistema operatiu MS-DOS i el Linux.

La **versió 1.0 del sistema operatiu MS-DOS** va ser introduïda el 1981 per una modesta companyia de programari anomenada Microsoft. L'MS-DOS era monousuari i amb força limitacions i problemes d'estabilitat deguts a la manca de dos modes d'execució (usuari i privilegiat) del maquinari en el qual s'executava (el processador 8088 d'Intel). L'MS-DOS va anar evolucionant adaptant-se als nous dispositius d'emmagatzemament i incorporant noves prestacions. Cap al 1990 va aparèixer el Windows 3.0, una interfície gràfica per a l'MS-DOS, però

el sistema operatiu continuava tenint força problemes d'estabilitat (la fatídica pantalla blava). Cap al 1993 va aparèixer la línia Windows-NT, seguida del Windows XP (2001), Windows Vista (2006) i Windows 7 (2009). Aquests darrers sistemes ja eren multiprocés i multiusuari, i van aconseguir assolir graus molt alts d'estabilitat perquè aprofitaven les característiques avançades dels processadors Intel. Per poder desenvolupar aplicacions portables entre les darreres versions del Windows, Microsoft ha definit una interfície estàndard anomenada Win32.

El **Linux** va néixer el 1991 quan un estudiant d'informàtica finès, Linus Torvalds, va decidir implementar una versió de l'UNIX que seguís l'estàndard POSIX, per a processadors 80386. Linus va utilitzar Internet per a buscar una comunitat de voluntaris que l'ajudés a mantenir i millorar el sistema operatiu. El Linux s'ofereix amb una llicència de programari lliure, de manera que el codi font està disponible per a tothom. Ara bé, per a fer usable el sistema operatiu cal disposar de programari de sistema (intèrprets d'ordres, compiladors, editors, etc.). Les distribucions del Linux (Debian, Red Hat, Fedora, Knoppix, Ubuntu, etc.) estan acompanyades per les aplicacions desenvolupades per la comunitat GNU. Per tant, alguns parlen de distribucions GNU/Linux per referir-se a tot el programari de sistema utilitzat a les instal·lacions del Linux. Aquest sistema està en evolució constant, adaptant-se a noves arquitectures, i cada cop és més amigable per a usuaris no experts i per a usuaris acostumats a sistemes Windows.

3. Els serveis que ofereix el sistema operatiu

El sistema operatiu ofereix una gran varietat de serveis que anirem ampliant en els mòduls següents.

Si ens fixem en la seva funció, els serveis es poden classificar en gestió de processos, senyalització de processos, gestió de dispositius d'entrada/sortida, gestió directa dels recursos del sistema, gestió del sistema d'arxius, proteccions i funcions de temps. A continuació descrivim algunes d'aquestes classes:

1) **Gestió de processos:** el concepte de procés és un dels més importants de l'assignatura, engloba tots els recursos necessaris per a executar un programa. En aquest apartat englobem els serveis que fan referència a la creació i eliminació de processos. Però també hi podem trobar altres serveis, com la suspensió i la reanimació de processos. També podem modificar atributs d'execució, com és la prioritat o el temps de quota de CPU assignat.

2) **Senyalització entre processos:** en un sistema multiprocés, el nexa natural d'unió entre tots els processos és el sistema operatiu. Aquest ofereix serveis per a comunicar i sincronitzar processos.

3) **Gestió de dispositius d'entrada/sortida:** els serveis relacionats amb el sistema d'entrada/sortida tenen com a funcions principals crear, obrir i tancar canals d'entrada/sortida, i també escriure'ls i llegir-los.

4) **Gestió directa dels recursos del sistema:** aquest grup de serveis és molt específic de cada sistema operatiu, però un exemple comú és la gestió de la memòria. Algunes de les operacions que s'efectuen amb la memòria del sistema són assignar, alliberar, protegir, compartir, etc.

5) **Gestió del sistema d'arxius:** juntament amb el bloc de serveis d'entrada/sortida, també és un grup de serveis molt utilitzat. La gestió del sistema de fitxers inclou les funcionalitats com crear i eliminar fitxers i directoris, canviar de directori, etc.

Per a poder sol·licitar aquests serveis cal utilitzar les crides al sistema que ofereix el sistema operatiu. Ara bé, tot i que les crides al sistema resulten un mecanisme molt còmode per als usuaris programadors, no és així per als usuaris finals del sistema operatiu. Per tant, mentre que els usuaris programadors sol·licitaran aquests serveis directament utilitzant crides al sistema, els usuaris finals sol·licitaran aquests serveis de manera indirecta, utilitzant l'interpret d'ordres o d'altres aplicacions de sistema.

1) Les **crides al sistema** permeten fer peticions directes al sistema operatiu. Les crides al sistema són la visió que té un programador dels serveis que pot oferir el sistema operatiu. Els llenguatges d'alt nivell en última instància fan una crida al sistema.

2) A un nivell més pròxim a l'usuari, el programari de sistema ofereix un programa, l'anomenat **intèrpret d'ordres**, gràcies al qual l'usuari pot dialogar amb el sistema operatiu sense necessitat d'escriure cap programa.

3.1. Les crides al sistema

Les crides al sistema (*system calls*) ofereixen les funcions bàsiques per a poder utilitzar tots els recursos del sistema de manera correcta i controlada.

Vegeu també

En la figura 1, apartat 1, les crides al sistema correspondrien als diversos punts d'entrada a la caixa del sistema operatiu.

Des del punt de vista del programador, una crida al sistema no és més que una rutina amb una sèrie de paràmetres d'entrada, una sèrie de paràmetres de sortida i una semàntica associada. Cada sistema operatiu ofereix un repertori de crides al sistema (com a referència, la versió 2.6.32 del sistema operatiu Linux ofereix 337 crides al sistema). Aquest repertori de crides al sistema constitueix l'API (*application programming interface*) del sistema operatiu.

Els dissenyadors de cada sistema operatiu han decidit quin repertori de crides al sistema ofereixen i quina semàntica hi associen. Comparant el repertori de crides al sistema de diferents sistemes operatius, podem trobar semblances i diferències significatives.

Tant l'UNIX com el Win32 ofereixen una crida al sistema per llegir dades d'un canal d'E/S. En el cas de l'UNIX, la interfície és `ssize_t read(int fd, void *buf, size_t count)`; en el cas del Win32 la interfície és `BOOL ReadFile(HANDLE hFile, LPVOID lpBuffer, DWORD nNumberOfBytesToRead, LPDWORD lpNumberOfBytesRead, LPOVERLAPPED lpOverlapped)`.

En canvi, la crida al sistema UNIX que permet crear un procés té la interfície `pid_t fork(void)`; mentre que l'equivalent Win32 té la interfície `BOOL CreateProcess(LPCTSTR lpApplicationName, LPTSTR lpCommandLine, LPSECURITY_ATTRIBUTES lpProcessAttributes, LPSECURITY_ATTRIBUTES lpThreadAttributes, BOOL bInheritHandles, DWORD dwCreationFlags, LPVOID lpEnvironment, LPCTSTR lpCurrentDirectory, LPSTARTUPINFO lpStartupInfo, LPPROCESS_INFORMATION lpProcessInformation)`.

3.2. L'intèrpret d'ordres

L'intèrpret d'ordres (o *shell*) és un programa encarregat d'interpretar i comunicar al SO el que vol fer l'usuari del sistema. Aquest programa reconeix un conjunt limitat d'ordres que, bàsicament, permeten a l'usuari executar programes i accedir, modificar, crear i protegir la informació. S'implementa mitjançant crides al sistema.

Vegeu també

En la figura 1, apartat 1, l'intèrpret d'ordres correspondria a una de les aplicacions que trobem dins del programari de sistema.

Mentre els terminals no disposaven de prestacions gràfiques, els intèrprets d'ordres treballaven en mode text; actualment, també hi ha intèrprets d'ordres que treballen en mode gràfic.



Figura 7. Intèrpret d'ordres en mode text

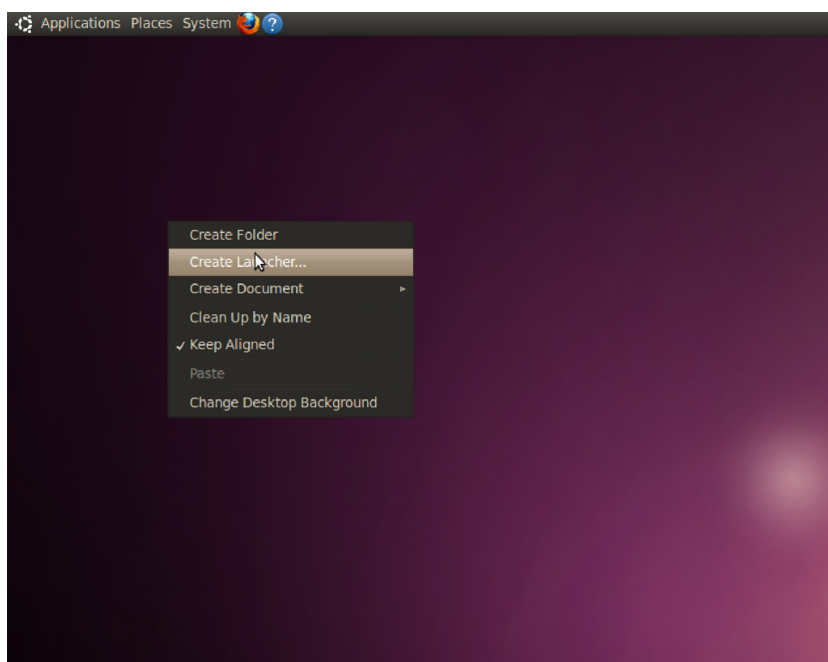


Figura 8. Intèrpret d'ordres en mode gràfic

Resum

Els sistemes operatius han evolucionat durant les darreres dècades, orientats cap a dos objectius principals:

- 1) Gestionar de manera eficient els recursos del sistema computador.
- 2) Proporcionar un entorn que faciliti el desenvolupament i l'execució de programes. Inicialment es treballava amb llenguatge màquina, però l'aparició de llenguatges d'alt nivell, compiladors i carregadors han facilitat la tasca de programació.

D'altra banda, l'evolució del maquinari (dispositius cada cop més ràpids) determina en cada moment els mètodes de treball i les tècniques que es desenvolupen.

Inicialment hi havia els sistemes per lots (*batch*) i les operacions fora de línia (*off-line*) que permetien encavalcar operacions d'entrada/sortida amb operacions de càlcul. Amb els sistemes operatius multiprogramats, diversos programes es carreguen en la memòria i la CPU en fa una execució concurrent per intentar aconseguir estar sempre ocupada. Finalment, tenim el temps compartit, que fa que diversos usuaris puguin utilitzar un computador interactivament al mateix temps.

Altres sistemes operatius inclouen els conceptes de xarxa, distribució de recursos i temps real.

Finalment, hem introduït els serveis que sol oferir un sistema operatiu. Aquests serveis es poden sol·licitar utilitzant crides al sistema. Per facilitar la feina als usuaris finals del sistema operatiu, dins del programari de sistema trobem una aplicació, l'interpret d'ordres, que amagarà a l'usuari final la complexitat inherent a les crides al sistema.

Activitats

1. Quines són les dues funcions principals d'un sistema operatiu?
2. Què és la multiprogramació? Quin n'és l'avantatge principal?
3. El concepte de multiprogramació té sentit en sistemes operatius en temps real? Per què? I el concepte de temps compartit? Per què?
4. Busqueu a Internet més informació sobre les diverses versions de l'MS-DOS/Windows.
5. Busqueu a Internet més informació sobre la història de l'UNIX i del Linux. Intenteu obtenir un diagrama en què apareguin els sistemes operatius de la família UNIX.
6. Busqueu a Internet el missatge que Linus Torvalds va enviar el 25 d'agost de 1991 anunciant que estava treballant en un sistema operatiu (Linux). Llegiu les reaccions que va provocar.
7. Busqueu a Internet informació sobre els sistemes operatius utilitzats pels ordinadors Apple.

Glossari

buffering Vegeu **emmagatzematge a la memòria intermèdia**.

emmagatzematge a la memòria intermèdia *m* Emmagatzematge que permet encavalcar el processament d'un treball amb la seva pròpia entrada/sortida.

en buffering

gestió de cues *f* Gestió que permet encavalcar l'entrada/sortida d'un treball amb l'entrada/sortida d'un altre.

en spooling

programari de sistema *m* Programari que permet utilitzar i gestionar el computador. Comprèn el sistema operatiu (nucli o *kernel*), que s'executa en mode d'execució privilegiat, i les biblioteques, els editors, l'interpret d'ordres, els compiladors, etc., que s'executen en mode d'execució no privilegiat (usuari).

sistema multiprogramat *m* Sistema que permet que diversos processos comparteixin el processador. Quan un procés deixa d'executar-se (en finalitzar o bé en fer una petició d'entrada/sortida) el processador passa a executar un altre procés.

sistema monousuari *m* Sistema en el qual el tipus usuari no està definit i, per tant, les dades que té el sistema són accessibles per a qualsevol que es pugui connectar.

sistema multiusuari *m* Sistema que distingeix entre diferents usuaris (grups d'usuaris) i permet controlar l'accés a la informació per part dels diferents grups d'usuaris que es puguin definir.

sistema operatiu distribuït *m* Sistema que està connectat en Xarxa que permet la compartició de recursos. Un usuari no ha de saber necessàriament on s'executen els seus processos o quin recurs concret de la Xarxa està utilitzant. El sistema gestiona tots els recursos de manera transparent per a l'usuari.

sistema operatiu en temps real *m* Sistema operatiu que ha de ser capaç de reaccionar en un temps màxim determinat davant esdeveniments externs.

sistema operatiu en xarxa *m* Sistema que està connectat per una xarxa d'interconnexió. L'usuari sap en tot moment en quina màquina de la xarxa s'executen els seus processos.

sistema per lots *m* Manera d'utilitzar un computador en què l'execució dels treballs no és interactiva. L'execució dels treballs sol ser seqüencial.

spooling Vegeu **gestió de cues**.

temps compartit *m* Sistema en què diversos processos poden compartir el processador. En aquest cas, cada procés té assignat el processador per un temps màxim, un quàntum. El processador és assignat a un altre procés quan el procés actualment en execució arriba al final de la seva execució, fa una operació d'entrada/sortida o esgota el seu quàntum.

treball fora de línia *m* Treball que s'efectua en un dispositiu fora de línia. Un dispositiu és fora de línia si està desconnectat del computador i, per tant, no pot dur a terme cap tipus de transferència directa.

Bibliografia

Silberschatz, A.; Galvin; Gagne, G. (2008). *Operating Systems Concepts*. (8a. ed.). John Wiley & Sons.

Tanenbaum, A. (2009). *Modern Operating Systems*. Prentice Hall.

