

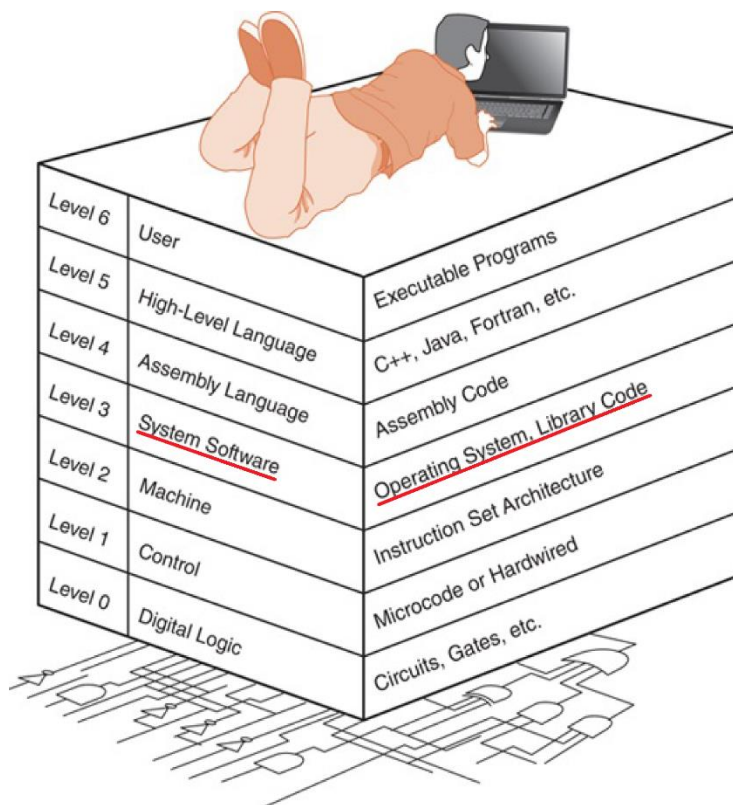


Kolegji UBT

Sistemet Operative Ligjërata 1

Hyrje

MSc. Valdrin Haxhiu



Hyrje

Sistemi Operativ është një softuer i cilin menaxhon harduerin e një kompjuteri. Sistemi operativ ofron një bazë për programet aplikative dhe luan rolin e një ndërmjetësuesi në mes të shfrytëzuesit të kompjuterit dhe harduerit të kompjuterit. Një aspekt interesant i sistemeve operative është se si ato dallojnë në përmbushjen e këtyre detyrave në një numër të madh të ambienteve kompjuterike. Sistemet operative janë kudo, që nga veturat e deri te pajisjet shtëpiake që përfshijnë pajisjet e “Internetit të gjërave” (Internet of things), telefonat e mençur (smartphones), kompjuterët personal, kompjuterët e ndërmarrjeve dhe ambientet kompjuterike në cloud.

Për të mësuar për rolin e një sistemi operativ në një ambient kompjuterik modern, është me rëndësi që së pari të kuptohet arkitektura dhe organizimi i harduerit të kompjuterit. Kjo përfshinë CPU-në, memorien, pajisjet hyrëse/dalëse dhe mediumet me kapacitet të madh për ruajtje. Një përgjegjësi themelore e një sistemi operativ është dhënia (ose ndarja) e këtyre resurseve për programe.

Meqë një sistem operativ është një sistem i madh dhe kompleks, duhet patjetër të ndërtohet nga pjesë më të vogla. Secila prej këtyre pjesëve duhet të jetë një anë e përcaktuar mirë e sistemit, me hyrje, dalje dhe funksione të definuara mirë. Në këtë ligjëratë, do të ofrojmë një trajtim të përgjithshëm të pjesëve kryesore të një sistemi kompjuterik modern si dhe të funksioneve që i ofron sistemi operativ.

1.1 Punët që i kryejnë sistemet operative

Diskutimin do të fillojmë duke trajtuar rolin e sistemit operativ brenda sistemit kompjuterik. Një sistem kompjuterik mund të ndahet në katër pjesë: **hardueri**, **sistemi operativ**, **programet aplikative** dhe **shfrytëzuesi** (Figura 1.1).

Hardueri – njësia qendrore përpunuese (CPU), memoria dhe pajisjet hyrëse/dalëse – ofron resurset bazë kompjuterike për sistemin. **Programet aplikative** – si programet për përpunimin e teksteve, për llogaritje, kompajlerët dhe ueb shfletuesit – përcaktojnë mënyrat se si këto resurse përdoren për të zgjidhur probleme të shfrytëzuesit me anë të kompjuterit. Sistemi operativ kontrollon harduerin dhe kordinon përdorimin e tij në mes programeve të ndryshme aplikative për shfrytëzues të ndryshëm.

Ne mund të trajtojmë një sistem kompjuterik si të përbërë prej harduerit, softuerit dhe të dhënave. Sistemi operativ ofron mundësinë e përdorimit në mënyrën e duhur të këtyre resurseve në punën e sistemit kompjuterik. Një sistem operativ është i ngjashëm me një qeveri. Sikur qeveria, vet sistemi operativ nuk kryen ndonjë funksion të dobishëm. Ai thjeshtë ofron një **ambient** brenda të cilit programet tjerë mund të kryejnë punë të dobishme.

Për të kuptuar në mënyrë më të plotë rolin e sistemit operativ, në vazhdim do të trajtojmë sistemet operative nga dy pikëvështrime: pikëvështrimi i shfrytëzuesit dhe i sistemit.

1.1 Pikëvështrimi i shfrytëzuesit

Pikëvështrimi i shfrytëzuesit për kompjuterin ndryshon varësisht prej interfejsit që përdoret. Shumica e shfrytëzuesve përdorin një laptop ose një kompjuter desktop që ka një monitor, tastierë dhe një maus. Sistemet si këto janë dizajnuar për një shfrytëzues i cili mund të përdorë

resurset vetëm për vete (mund të monopolizojë). Qëllimi është të maksimizohet puna (ose loja) të cilën e kryen shfrytëzuesi. Në këtë rast, sistemi operativ është dizajnuar kryesisht për **lehtësi në përdorim**, me një vëmendje kushtuar performancës, sigurisë dhe aspak kujdes për **përdorimin e resurseve** – mënyra se si resurse të ndryshme harduerike dhe softuerike përdoren nga shumë shfrytëzues.

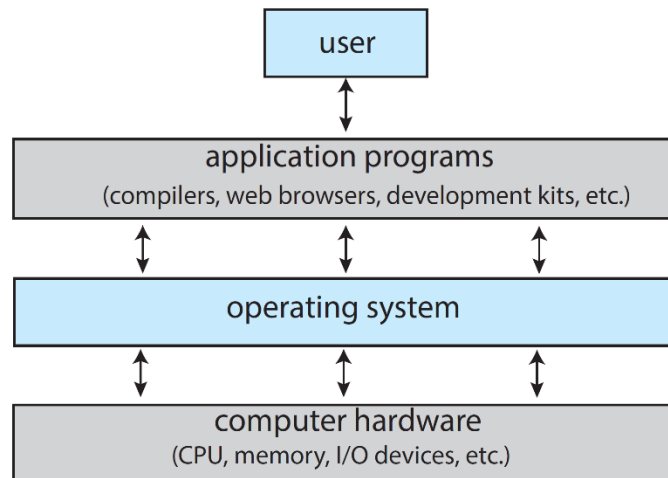


Figura 1.1 Një pamje e thjeshtë e pjesëve të një sistemi kompjuterik

Gjithnjë e në rritje, shumë shfrytëzues përdorin pajisje mobile si telefonë të mençur dhe tabletë – pajisje këto të cilat për disa shfrytëzues janë duke i zëvendësuar sistemet kompjuterike desktop dhe laptop. Këto pajisje zakonisht janë të lidhura në rrjete celulare ose me ndonjë teknologji tjetër pa tela. Interfejsi i shfrytëzuesit për kompjuterët mobil në përgjithësi ka një **ekran me prekje**, ku shfrytëzuesi komunikon me sistemin duke shtypur ose rrëshqitur gishtërinjtë përgjatë ekranit në vend se të përdorin një tastierë fizike ose maus. Shumë pajisje mobile lejojnë shfrytëzuesit të përdorin edhe mundësinë e **njohjes së zërit (voice recognition)** si interfejs, sikur mundësia **Siri** nga Apple.

Disa kompjuterë kanë pak ose fare mundësi që të shihen nga shfrytëzuesi. Për shembull, **kompjuterët embedded** (kompjuter brenda një sistemi më të madh) në pajisje shtëpiake dhe vetura mund të kenë tastiera numerike dhe mund të ndezin ose ndalin drita treguese për të treguar gjendjen, por ato, sistemet e tyre operative dhe aplikacionet janë dizajnuar të përdoren kryesisht pa intervenimin e shfrytëzuesit.

1.2 Pikëvështrimi i sistemit

Nga pikëvështrimi i kompjuterit, sistemi operativ është programi i cili është i lidhur ngushtë me harduerin. Në këtë kontekst, sistemin operativ mund të trajtojmë si një **dhënës të resurseve**. Një sistem kompjuterik ka shumë resurse që mund të nevojiten për zgjidhjen e një problemi: kohën e CPU-së, hapësirën e memories DRAM, hapësirën në memoriet me kapacitet të madh, pajisjet hyrëse/dalëse e të tjera. Sistemi operativ vepron si menaxhues i këtyre resurseve. Duke u përballë me shumë kërkesa dhe me mundësinë që ato të jenë në konflikt me njëra-tjetrën, sistemi operativ duhet patjetër të vendosë se si të japë (ndajë) resurset për programet dhe shfrytëzuesit e caktuar ashtu që të mund të punojë me sistemin kompjuterik në mënyrë efikënte dhe të drejtë.

Një pikëvështrim pak më ndryshe i një sistemi operativ thekson nevojën për të kontrolluar pajisje të ndryshme hyrëse/dalëse dhe programe të shfrytëzuesit. Sistemi operativ është një program për kontroll. Një **program kontrolli** menaxhon ekzekutimin e programeve të shfrytëzuesit për të parandaluar gabimet dhe përdorimin jo të duhur të kompjuterit. Merret posaçërisht me punën dhe kontrollin e pajisjeve hyrëse/dalëse.

1.3 Definimi i sistemeve operative

Tani për tani, mund të vëreni që termi *sistem operativ* mbulon shumë role dhe funksione. Është ashtu, së paku pjesërisht, për arsye të një numri shumë të madh të dizajnëve dhe përdorimit të kompjuterëve. Kompjuterët janë prezentë brenda tosterëve, veturave, anijeve, anijeve kozmike, shtëpive dhe bizneseve. Ata paraqesin bazën për makinat e lojërave, sistemet kabllorë për TV dhe sistemet e kontrollit industrial.

Për të shpjeguar këtë llojllojshmëri, mund t'i kthehemi historisë së kompjuterëve. Edhe pse kompjuterët kanë një histori relativisht të shkurtër, ata kanë evoluar shumë shpejtë. Kompjuterika filloi si një eksperiment për të përcaktuar së çfarë mund të bëhej dhe shpejtë shkoi te sistemet me vetëm një qëllim përdorimi për përdorim nga ushtria, si thyerja e kodeve sekrete dhe llogaritja e trajektoreve të armëve dhe përdorimi qeveritar, si përpunimi i statistikave të zgjedhjeve. Ata kompjuterë të hershëm evoluan në kompjuterë mainframe me qëllim të përgjithshëm dhe me shumë funksione, në atë kohë lindën sistemet operative. Kompjuterët u pasuruan me më shumë funksione dhe u zvogëlua madhësia e tyre, duke bërë të mundur mënyra të shumta përdorimi dhe një numër të madh e të ndryshëm të sistemeve operative.

Atëherë, si mund të definojmë se çka është një sistem operativ? Në përgjithësi, nuk ka ndonjë definim të plotë adekuat për një sistem operativ. Sistemet operative ekzistojnë sepse ato ofrojnë një mënyrë të arsyeshme për zgjidhjen e problemit të bërjes së një sistemi kompjuterik të shfrytëzueshëm. Qëllimi themelor i sistemeve kompjuterike është ekzekutimi i programeve dhe të bëjnë të lehtë zgjidhjen e problemeve të shfrytëzuesit. Hardueri i kompjuterit është i ndërtuar drejtë këtij qëllimi. Meqë hardueri si i vetëm nuk është edhe aq i lehtë për përdorim, zhvillohen programe aplikative për të bërë më të lehtë përdorimin. Këto programe kanë nevojë për operacione si kontrollimi i pajisjeve hyrëse/dalëse. Funksionet si kontrollimi dhe dhënia e resurseve bëhen bashkë në një program: sistemi operativ.

Përveç kësaj, nuk ka ndonjë definicion të pranueshëm në përgjithësi se çfarë është pjesë e një sistemi operativ. Një vështrim i thjeshtë është vështrimi se sistemi operativ përfshinë çdo gjë që vjen me të kur ne blejmë “sistemin operativ”. Karakteristikat e përfshira, megjithatë, dallojnë shumë në sisteme të ndryshme. Disa sisteme harxhojnë më pak se 1 MB hapësirë në memorie dhe nuk kanë editor që mund të përdoret lehtë, përderisa disa sisteme tjera kërkojnë disa GB hapësirë të memories dhe janë të bazuara plotësisht në sisteme përmes interfejsit grafik. Një definim më i zakonshëm, definimi që përdorim më shpesh, është ai që sistemi operativ është programi që ekzekutohet gjatë tërë kohës (gjatë kohës kur kompjuteri është duke punuar) në kompjuter – zakonisht i njohur si **kerneli (bërthama ose pjesa kryesore)**. Krahas kernelit, janë edhe dy lloje të softuerëve: **programet e sistemit**, që janë pjesë e sistemit operativ, por që nuk janë domosdosmërisht pjesë e kernelit, dhe programet aplikative, që përfshijnë të gjitha programet që nuk janë pjesë e sistemit operativ.

Çështja se çka është pjesë e sistemit operativ u bë gjithnjë e më e rëndësishme kur kompjuterët personal u përhapën më shumë dhe sistemet operative u bënë më të sofistikuar. Në vitin 1998, Departamenti i Drejtësisë i Shteteve të Bashkuara paditi në gjykatë Microsoft-in, duke

u ankuar se Microsoft-i kishte përfshirë shumë funksione brenda sistemeve të tij operative dhe në atë mënyrë kishte pamundësuar konkurrencën në mes kompanive që zhvillojnë programe aplikative. (Për shembull, një ueb shfletues ishte integruar brenda sistemeve operative të Microsoft-it). Rezultati i padisë në gjykatë ishte se Microsoft-i doli fajtorë për përdorimin e monopolit të sistemeve të tij operative për të kufizuar kompanitë konkurente.

Sot, megjithatë, nëse i shikojmë sistemet operative për pajisje mobile, shohim se përsëri ka një rritje të programeve që përfshihen në sistem operativ. Sistemet operative të pajisjeve mobile zakonisht përfshijnë jo vetëm një kernel por edhe **middleware** – një bashkësi e kornizave (frameworks) për zhvillim të softuerit që ofrojnë shërbime shtesë për zhvilluesit e aplikacioneve. Për shembull, secili nga dy prej sistemeve operative mobile më të njohura, - iOS nga Apple dhe Android nga Google – përfshinë një kernel krahas middleware që përkrahë bazat e të dhënave, multimedia dhe grafikë (këto janë vetëm disa prej tyre).

Si përmbledhje, për qëllimet tona, sistemi operativ përshinë kernelin që ekzekutohet gjatë tërë kohës, kornizat middleware që lehtësojnë zhvillimin e aplikacioneve, ofrojnë karakteristika dhe programe të sistemit që ndihmojnë në menaxhimin e sistemit gjatë kohës kur është duke punuar. Pjesa më e madhe e këtij libri merret me kernelin e sistemeve operative për qëllime të përgjithshme, por merret edhe me pjesë tjera që nevojiten për të trajtuar në mënyrë më të plotë dizajnin dhe punën e sistemeve operative.

1.4 Punët që i kryen sistemi operativ

Një sistem operativ ofron ambientin në të cilin ekzekutohen programet. Nga brenda tyre, sistemet operative ndryshojnë shumë, meqë janë të organizuara përgjatë vijave të ndryshme të dizajnit. Megjithatë, kanë shumë gjëra të përbashkëta që do t'i trajtojmë në këtë pjesë.

Që një kompjuter të mund të punojë – për shembull, kur të lëshohet në punë ose ristartohet – kompjuteri ka nevojë për një program fillestar që të fillojë punën. Programi fillues i njohur si programi **bootstrap** zakonisht është i thjeshtë. Zakonisht ruhet brenda harduerit në **firmware**. Fillon të gjitha pjesët e sistemit kompjuterik, që nga regjistrat e CPU-së te kontrolluesit e pajisjeve e deri te përmbajtjet e memories. Programi bootstrap duhet patjetër të jetë në gjendje të vendosë sistemin operativ në memorien DRAM dhe të fillojë punën e sistemit. Për të realizuar këtë punë, programi bootstrap duhet të gjejë kernelin e sistemit operativ (në disk SSD ose HDD) dhe të vendosë atë në memorien DRAM.

Pasi kerneli të jetë vendosur në memorien DRAM dhe është duke u ekzekutuar, mund të të ofrojë shërbime për sistemin dhe shfrytëzuesit e tij. Disa nga shërbimet ofrohen jashtë kernelit nga programet e sistemit të cilat vendosen në memorien DRAM gjatë kohës së fillimit të punës të sistemit kompjuterik për t'u bërë programe të njohura si programe **daemon**, të cilat ekzekutohen gjatë tërë kohës kur është duke u ekzekutuar kerneli. Në sistemin operativ Linux, programi i parë i sistemit është programi “systemd,” i cili fillon shumë programe tjerë daemon. Pas përfundimit të kësaj faze, sistemi ka filluar punën në tërësi dhe sistemi pret për paraqitjen e ndonjë ngjarjeje.

Nëse nuk ka ndonjë proces për ekzekutim, asnjë pajisje hyrëse/dalëse për t'u shërbyer, asnjë shfrytëzues për t'u përgjigjur, sistemi operativ qëndron në pritje për ndonjë ngjarje në sistem kompjuterik. Paraqitja e ndonjë ngjarjeje sinjalizohet përmes ndonjë ndërprerjeje (interrupt). Një lloj tjetër i ndërprerjes është ndërprerja **trap** (ose një **exception**), e cila është një ndërprerje e shkaktuar nga softueri nga ndonjë gabim (për shembull, pjesëtimi me zero ose qasje jo valide në memorie) ose ndonjë kërkesë specifike e një programi të shfrytëzuesit që një

shërbim i sistemit operativ të kryhet duke ekzekutuar një operacion të veçantë të njohur si **system call (thirrje sistmore)**.

1.4.1 Multiprogramimi dhe multitasking

Njëri nga aspektet më të rëndësishme të sistemeve operative është aftësia për të ekzekutuar shumë programe, meqë një program i vetëm, në përgjithësi, nuk mund të mbajë të zënë gjatë tërë kohës CPU-në ose pajisjet dalëse. Për më tepër, shfrytëzuesit zakonisht *duan* të ekzekutojnë më shumë se një program në të njëjtën kohë. **Multiprogramimi** e rritë përdorimin e CPU-së, duke i bërë të lumtur edhe shfrytëzuesit, me organizimin e ekzekutimit të programeve ashtu që CPU në çdo kohë ka një program për ekzekutim. Në një sistem me multiprogramim, një program që është duke u ekzekutuar njihet si **proces**.

Ideja qëndron kështu: sistemi operativ i mban disa procese në memorien DRAM në të njëjtën kohë (Figura 1.2).

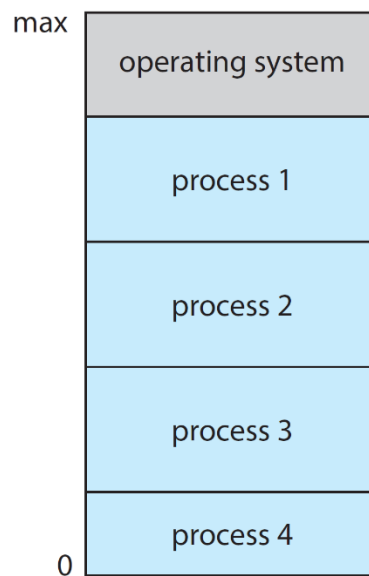


Figura 1.2 Shtrirja e memories për një sistem multiprogramimi

Sistemi operativ përzgjedhë dhe fillon të ekzekutojë njërin nga këta procese. Ndërkohë, procesi mund të ketë nevojë të presë për ndonjë punë, përfundimin e një operacioni të hyrje/daljes. Në një sistem pa multiprogramim, CPU-ja thjeshtë nuk do të kryente ndonjë punë tjetër. Në një sistem me multiprogramim, sistemi operativ thjeshtë përzgjedhë një proces tjetër për ekzekutim. Kur edhe **ai** proces ka nevojë të presë, CPU-ja përzgjedhë një proces **tjetër** për ekzekutim. Ndërkohë, procesi i parë përfundon pritjen dhe merr për vete përsëri CPU-në. Për sa kohë që së paku një proces ka nevojë për ekzekutim, CPU nuk rrin duke mos bërë asgjë.

Kjo ide është e zakonshme edhe në situata tjera jetësore. Për shembull, një avokate nuk punon me vetëm një klient. Përdërisa njëri rast është duke pritur që të shkojë në gjykatë ose të shtypen dokumentet, avokatja mund të punojë për ndonjë klient tjetër. Nëse ajo ka mjaftueshëm klientë, avokatja asnjëherë nuk do të rrinte duke mos bërë asgjë në mungesë të punëve.

Multitasking është një zgjerim i natyrshëm i multiprogramimit. Në sistemet me multitasking, CPU ekzekuton shumë procese duke kaluar nëpër to, por kalimi bëhet shpesh, duke i ofruar shfrytëzuesit një **kohë reagimi** të shkurtër. Kemi parasysh që kur një proces ekzekutohet, zakonisht procesi ekzekutohet për një kohë të shkurtër kur të përfundojë plotësisht

ose ka nevojë për ndonjë operacion hyrjeje/daljeje. Hyrje/dalja mund të jetë interaktive; në kuptimin, dalja paraqitet në një ekran për shfrytëzuesin dhe hyrja vjen nga një tastierë e shfrytëzuesit, maus ose ekran me prekje. Meqë hyrja/dalja interaktive varet nga shpejtësia e punës së shfrytëzuesit, mund të marrë kohë të gjatë deri në përfundim. Për shembull, hyrja varet nga shpejtësia me të cilën shkruan shfrytëzuesi në tastierë; shtatë karaktere për një sekondë janë shpejtë për njerëzit, por ngadalë për kompjuterët. Në vend se CPU të presë duke mos bërë asgjë në kohën kur ndodhë një hyrje interaktive, sistemi operativ do të përzgjedhë shpejtë për ekzekutim një proces tjetër.

Ekzistenca në të njëjtën kohë e disa proceseve në memorien DRAM kërkon një mënyrë të menaxhimit të memories. Për më tepër, nëse disa procese janë gati për ekzekutim në të njëjtën kohë, sistemi duhet të përzgjedhë se cili proces do të ekzekutohet në vazhdim. Kjo punë paraqet planifikim të kohës së CPU-së. Në fund, ekzekutimi i shumë proceseve në mënyrë konkurente kërkon që mundësia që proceset të ndikojnë në njëri-tjetrin të kufizohet në të gjitha fazat e sistemit operativ, përfshirë planifikimin e proceseve, ruajtjen në disk dhe menaxhimin e memories.

Në një sistem multiprogramimi, sistemi operativ duhet të sigurojë një kohë reagimi të arsyeshme. Një metodë e zakonshme për të bërë këtë është **memoria virtuale**, një teknikë që lejon ekzekutimin e një procesi që nuk është i vendosur plotësisht në memorien DRAM. Përparësia kryesore e kësaj skeme është se mundëson shfrytëzuesit të ekzekutojnë programe që janë më të mëdha se sa **memoria fizike (memoria DRAM)**. Për më tepër, memoria virtuale e paraqet memorien DRAM si një hapësirë të vazhdueshme të vendeve në memorie, duke e ndarë **memorien logjike** që shihet nga shfrytëzuesi prej memories fizike. Ky organizim i liron programuesit nga brenga për kufizimet e madhësisë së memories.

Sistemet me multiprogramim dhe multitasking duhet patjetër të ofrojnë një sistem të fajllave (file system). Sistemi i fajllave ndodhet në një memorie me kapacitet të madh; në atë mënyrë, duhet patjetër të ofrohet menaxhim i ruajtjes. Sistemi duhet patjetër të ofrojë edhe mbrojtje të resurseve nga përdorimi jo i duhur. Për të siguruar ekzekutim të rregullt, sistemi duhet patjetër të ofrojë mekanizma për sinkronizim dhe komunikim në mes të proceseve, e të sigurojë që proceset të mos bllokohen në një situatë deadlock, duke pritur përgjithmonë për njëri-tjetrin.

1.4.2 Puna me dy mënyra dhe me shumë mënyra

Meqë sistemi operativ dhe shfrytëzuesit e tij i përdorin bashkë resurset harduerike dhe softuerike të sistemit kompjuterik, një sistem operativ i dizajnuar mirë duhet të sigurohet patjetër që një program me probleme (ose me qëllim të keq) nuk mund të bëjë që programet tjera ose edhe vetë sistemi operativ të ekzekutohen me probleme. Për të siguruar punën në mënyrën e duhur të sistemit, duhet të jemi në gjendje të dallojmë ekzekutimin e kodit të sistemit operativ dhe kodit të programeve aplikative të shfrytëzuesit. Qasja që përdoret në shumë sisteme kompjuterike është që të ofrohet përkrahje harduerike që lejon dallimin në mes të mënyrave të ndryshme të ekzekutimit.

Do të kishim nevojë për së paku dy **mënyra** të ndara të punës: **mënyra e shfrytëzuesit (user mode)** dhe **mënyra e kernelit** (e njohur ndryshe edhe si **mënyra e mbikëqyrësit (supervisor mode)**, **system mode** ose **privileged mode**). Një bit, i quajtur si **biti i mënyrës (mode bit)**, i shtohet harduerit të kompjuterit për të treguar mënyrën aktuale të punës: kerneli (biti 0) ose shfrytëzuesi (biti 1). Përmes bitit të mënyrës, mund të dallojmë një punë që është duke u kryer për sistemin operativ dhe një punë që është duke u kryer për shfrytëzuesin. Kur

sistemi operativ është duke ekzekutuar një program për shfrytëzuesin, sistemi është në mënyrën e shfrytëzuesit. Megjithatë, kur një program i shfrytëzuesit kërkon një shërbim nga sistemi operativ (përmes një thirrjeje sistmore), sistemi duhet të kalojë nga mënyra e shfrytëzuesit në mënyrën kernel për të përmbushur kërkesën. Kjo është paraqitur në Figura 1.3. Siç do të shohim, ky përmirësim arkitekturor është i dobishëm edhe për shumë aspekte tjera të punës së sistemit.

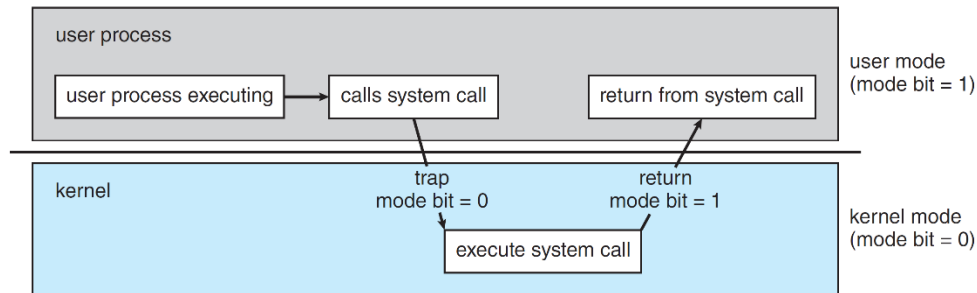


Figura 1.3 Kalimi prej mënyrës së shfrytëzuesit në mënyrën kernel

Në kohën e fillimit të punës të sistemit, hardueri fillon në mënyrën e kernelit. Në vazhdim, sistemi operativ vendoset në memorien DRAM dhe fillon programet e shfrytëzuesit në mënyrën e shfrytëzuesit. Kur të paraqitet ndonjë trap ose ndërprerje, hardueri kalon prej mënyrës së shfrytëzuesit në mënyrën e kernelit (e vendosë bitin e mënyrës me vlerën 0). Ashtu, sa herë që sistemi operativ e merr nën kontroll kompjuterin është në mënyrën e kernelit. Sistemi çdo herë kalon në mënyrën e shfrytëzuesit (duke vendosur vlerën 1 në bitin e mënyrës) para se t'i japë kontrollin një programi të shfrytëzuesit.

Puna me dy mënyra të punës ofron mundësinë e mbrojtjes së sistemit operativ nga shfrytëzuesit problematik – dhe shfrytëzuesit problematik nga njëri-tjetri. Kjo mbrojtje arrihet duke i trajtuar disa instruksione të cilat mund të shkaktojnë dëme (nëse keqpërdoren) si **instruksione të privileguara**. Hardueri lejon instruksionet e privileguara të ekzekutohen në mënyrën e kernelit. Nëse ka ndonjë tentim që një instruksion i privileguar të ekzekutohet në mënyrën e shfrytëzuesit, hardueri nuk e ekzekuton instruksionin, por e trajton si jo të lejueshëm dhe e kthen kontrollin te sistemi operativ.

Instruksioni për kalim në mënyrën e kernelit është një shembull i një instruksioni të privileguar. Shembuj të tjerë përfshijnë kontrollin e hyrje/daljes, menaxhimin e tajmerit dhe të ndërprerjeve.

Koncepti i mënyrave të punës mund të zgjerohet në më shumë se dy mënyra. Për shembull, procesorët e Intel-it kanë katër **unaza mbrojtëse** të ndara, ku unaza 0 është mënyra kernel dhe unaza 3 është mënyra e shfrytëzuesit. (Edhe pse unazat 1 dhe 2 do të mund të përdoren për shërbime të ndryshme të sistemit operativ, në praktikë përdoren rrallë). Sistemet ARMv8 kanë shtatë mënyra. CPU-të që përkrahin virtualizimin kanë një mënyrë të veçantë për të treguar atë mënyrë të punës kur **menaxhuesi i makinës virtuale (VMM)** e ka në kontroll sistemin. Në këtë mënyrë, VMM ka më shumë privilegje se proceset e shfrytëzuesit, por më pak se mënyra kernel. Ka nevojë për atë nivel të privilegjit që të mund të krijojë dhe menaxhojë makina virtuale, duke e ndërruar gjendjen e CPU-së për të kryer atë punë.

Mund të trajtojmë më shumë detaje të ekzekutimit të instruksioneve. Kontrolli fillestar i takon sistemit operativ, ku instruksionet ekzekutohen në mënyrën e kernelit. Kur kontrolli i jepet një programi të shfrytëzuesit, vendoset mënyra e shfrytëzuesit. Ndërkohë, kontrolli i kthehet sistemit operativ përmes një ndërprerjeje, trap ose thirrje sistmore. Shumica e sistemeve

operative moderne – si Microsoft Windows, UNIX dhe Linux – e shfrytëzojnë mundësinë e dy mënyrave të punës dhe ofrojnë mbrojtje më të madhe për sistemin operativ.

Thirrjet sistmore ofrojnë mundësinë që një program i shfrytëzuesit të kërkojë nga sistemi operativ që të kryejë ndonjë punë që mund të kryejë vetëm sistemi operativ për programin e shfrytëzuesit. Një thirrje sistmore realizohet në formë të ndryshme, varësisht prej funksionalitetit që ofron procesori. Të gjitha format janë metoda se si një proces mund të kërkojë ndonjë veprim nga sistemi operativ. Thirrja sistmore merr formën e një **trap** në një vend të caktuar në tabelën e vektorit të ndërprerjes. Një trap i tillë mund të ekzekutohet nga një instruksion i përgjithshëm trap, edhe pse disa sisteme kanë një instruksion specifik **syscall** për të realizuar një thirrje sistmore.

Kur një thirrje sistmore ekzekutohet, hardueri zakonisht e trajton si ndërprerje softuerike. Kontrolli bartet përmes vektorit të ndërprerjes të një procedurë në sistemin operativ dhe vendoset mënyra kernel. Procedura (funksioni) e cila merret me thirrjen sistmore është pjesë e sistemit operativ. Kernel e analizon instruksionin që ka shkaktuar ndërprerje për të përcaktuar cila thirrje sistmore është paraqitur; një parametër tregon se çfarë lloji të shërbimit kërkon programi i shfrytëzuesit. Informatat tjera shitesë të kërkesës mund të barten në regjistra, në stek ose në memorie (duke i vendosur në regjistra adresat e vendeve në memorie). Kerneli i vërteton parametrat se a janë mirë dhe të lejueshëm, ekzekuton kërkesën dhe kthen kontrollin te instruksioni që ndodhet pas instruksionit që ka shkaktuar thirrje sistmore.

Pasi mbrojtja nga hardueri të jetë prezente, mbrojtja i gjenë gabimet të cilat shkelin mënyrën e punës. Me këto gabime zakonisht merret sistemi operativ. Nëse një program i shfrytëzuesit dështon – duke bërë një tentim që të ekzekutojë një instruksion jo të lejueshëm ose tentim të qasjes në memorien DRAM në një hapësirë të adresave që nuk është e shfrytëzuesit, atëherë hardueri ia kthen kontrollin sistemit operativ. Forma trap e bartë kontrollin përmes vektorit të ndërprerjes të sistemit operativ, sikur një ndërprerje. Kur paraqitet një gabim në program, sistemi operativ duhet patjetër të ndalojë programin në mënyrë jo të rregullt. Kjo situatë trajtohet nga i njëjti kod që përdoret kur programi bën vet kërkesën për ndalje në mënyrë jo të rregullt.

1.4.3 Tajmeri

Duhet të sigurohemi që sistemi operativ e mirëmban kontrollin mbi CPU-në. Nuk duhet lejuar që një program i shfrytëzuesit të bllokohet në një unazë të pafundme ose të dështojë në thirrjen e shërbimeve të sistemit dhe të mos kthejë më kontrollin te sistemi operativ. Për të arritur këtë qëllim, mund të përdorim një **tajmer**. Një tajmer mund të vendoset që të ndërprejë punën aktuale në CPU pas një periode kohore të caktuar. Perioda mund të jetë konstante (për shembull, 1/60 e sekondës) ose e ndryshueshme (për shembull, nga 1 milisekondë deri në 1 sekondë). Një **tajmer i ndryshueshëm** zakonisht realizohet nga një orë me frekuencë konstante dhe një numërues. Sistemi operativ e vendosë vlerën në numërues. Në çdo puls të orës, vlera e numëruesit zvogëlohet për një. Kur numëruesi arrin vlerën 0, paraqitet një ndërprerje. Për shembull, një numërues 10 bitësh me një orë 1 milisekondëshe lejon ndërprerje në intervale prej 1 milisekondë deri në 1024 milisekonda, në hapa 1 milisekondësh.

Para se të kthejë kontrollin te shfrytëzuesi, sistemi operativ sigurohet që vlera e tajmerit të jetë vendosur për ndërprerje. Nëse tajmeri shkaktonte ndërprerje, kontrolli bartet automatikisht te sistemi operativ, që mund të trajtojë ndërprerjen si një gabim shumë të madh ose mund t'i japë

më shumë kohë programit. Sigurisht, instruksionet që ndërrojnë vlerën e tajmerit janë instruksione të privileguara.

1.5 Menaxhimi i resurseve

Sistemi operativ është një **menaxhues i resurseve**. CPU-ja e sistemit, hapësira e memories, hapësira për ruajtje të fajllave dhe pajisjet hyrëse/dalëse janë disa nga resurset të cilat sistemi operativ duhet patjetër t'i menaxhojë.

1.5.1 Menaxhimi i proceseve

Një program nuk mund të bëjë asgjë nëse instruksionet e tij nuk ekzekutohen në CPU. Një program që është duke u ekzekutuar është një proces. Një program si kompajleri është një proces dhe një program për përpunimin e teksteve që është duke u përdorë nga një shfrytëzues në një kompjuter është një proces. Njëjtë, një aplikacion për rrjete sociale në një pajisje mobile është një proces. Tani për tani, procesin mund të konsiderojmë si një kopje të një programi që është duke u ekzekutuar, por më vonë do të shohim se procesi është një koncept më i gjerë. Ka mundësi që të ofrohen thirrje sistimore që lejojnë që proceset të krijojnë nënprocese për ekzekutim konkurent.

Një proces ka nevojë për disa resurse – përfshirë kohën e CPU-së, memorien, fajllat dhe pajisjet hyrëse/dalëse – për të kryer punën e tij. Këto resurse i jepen zakonisht një procesi që është duke u ekzekutuar. Përveç disa resurseve fizike dhe logjike të cilat një proces i merr kur të krijohet, procesit mund t'i barten edhe disa të dhëna (hyrje) fillestare. Për shembull, le të marrim një proces që është duke ekzekutuar një ueb shfletues detyra e të cilit është të paraqesë një ueb faqe në një ekran. Procesit do t'i jepet një identifikues unik (URL) si hyrje dhe do të ekzekutojë instruksionet përkatëse dhe thirrjet sistimore për të marrë dhe paraqitur në ekran informacionin e dëshiruar. Kur procesi të përfundojë, sistemi operativ do t'i marrë përsëri resurset e ripërdorshme.

Vetë programi nuk është një proces. Një program është një entitet **pasiv**, sikur përmbajtjet e një fajlli të ruajtur në disk, ndërsa një proces është një entitet **aktiv**. Një proces me vetëm një thred ka një regjistër PC (program counter) për adresën e instruksionit të radhës për ekzekutim. Ekzekutimi i një procesi të tillë duhet të bëhet patjetër në mënyrë sekuenciale. CPU-ja ekzekuton secilin instruksion të procesit një nga një deri në përfundimin e procesit. Për më tepër, në një moment kohor, më së shumti ekzekutohet vetëm një instruksion i procesit. Kështu, edhe pse i njëjti program mund të ketë dy procese, ato procese konsiderohen si dy radhë të ndryshme të ekzekutimit. Një proces me shumë threda përdorë disa regjistra PC, ku secili ka adresën e instruksionit të radhës të thredit për ekzekutim.

Një proces është një njësi e punës në sistem. Një sistem përbëhet nga një bashkësi e proceseve, disa prej të cilave janë procese të sistemit operativ (ato që ekzekutojnë kodin e sistemit) dhe pjesa tjetër janë procese të shfrytëzuesit (ato që ekzekutojnë kodin e programeve të shfrytëzuesit). Të gjitha këto procese ka mundësi të ekzekutohen në mënyrë konkurente – duke përdorë të njëjtën bërthamë të CPU-së – ose në mënyrë paralele në disa bërthama të CPU-së.

Sistemi operativ është përgjegjës për këto aktivitete në lidhje me menaxhimin e proceseve:

- Krijimi dhe fshirja e proceseve të shfrytëzuesit dhe të sistemit
- Planifikimi për ekzekutim të proceseve dhe thredave në CPU
- Ndalja e përkohshme dhe vazhdimi i proceseve
- Ofrimi i mekanizmave për sinkronizim të proceseve
- Ofrimi i mekanizmave për komunikim në mes të proceseve

1.5.2 Menaxhimi i memories

Memoria ka një rëndësi të madhe në funksionimin e sistemit kompjuterik. Memoria DRAM është një varg i madh i bajtëve, memoria DRAM mund të jetë e madhësisë prej disa qindra mijëra deri në disa miliardë bajtë. Secili bajt ka adresën e tij. Memoria kyresore (memoria DRAM) është një vend për ruajtjen e të dhënave që mund të qasen (shkrim/lexim) shpejtë e mund të përdoren bashkë nga CPU dhe pajisjet hyrëse/dalëse. CPU-ja lexon instruksionet nga memoria DRAM gjatë fazës së leximit të instruksionit dhe lexon e shkruan të dhëna në memorien DRAM gjatë fazës së leximit të të dhënave (në arkitekturën von Neumann). Memoria DRAM është në përgjithësi memoria e madhe e vetme të cilën CPU-ja është në gjendje të adresojë dhe t'i qaset direkt. Për shembull, në mënyrë që CPU-ja të përpunojë të dhëna nga disku, ato të dhëna duhet patjetër që së pari të barten në memorien DRAM me anë të komandave të lëshuara nga CPU-ja për pajisje hyrëse/dalëse. Në të njëjtën mënyrë, instruksionet duhet patjetër të jenë në memorien DRAM që CPU-ja të mund t'i ekzekutojë ato.

Për të ekzekutuar një program, programi duhet patjetër të vendoset (pasqyrohet) përmes adresave fizike në memorien DRAM. Gjatë ekzekutimit të tij, programi lexon instruksionet dhe të dhënat në memorie përmes adresave fizike. Dikur programi përfundon ekzekutimin, hapësira e memories që kishte zënë më herët (kur ishte në memorie) lirohet dhe ndonjë program tjetër mund të vendoset aty dhe të ekzekutohet.

Për të përmirësuar shfrytëzimin e CPU-së dhe shpejtësinë e përgjigjes së kompjuterit ndaj shfrytëzuesve, kompjuterët për qëllime të përgjithshme duhet patjetër të mbajnë në memorie disa programe, duke paraqitur nevojën për menaxhim të memories. Përdoren disa skema të ndryshme të menaxhimit të memories. Këto skema reflektojnë qasje të ndryshme dhe efikasiteti i algoritmit varet prej situatës. Gjatë përzgjedhjes së një skeme për menaxhim të memories për ndonjë sistem specifik, duhet patjetër të kemi parasysh shumë faktorë – sidomos dizajnin *harduerik* të sistemit. Secili algoritm ka nevojë për përkrahje nga hardueri.

Sistemi operativ është përgjegjës për këto aktivitete në lidhje me menaxhimin e memories:

- Mbajtja e shënimeve se cilat pjesë të memories janë duke u përdorë në momentin e caktuar dhe cili proces është duke i përdorë ato
- Dhënia dhe marrja e hapësirës së memories sipas nevojës
- Marrja e vendimit se cilat procese (ose pjesë të proceseve) dhe të dhëna duhet të vendosen dhe të largohen nga memoria

1.5.3 Menaxhimi i sistemit të fajllave

Për të bërë sa më të lehtë për përdorim sistemin kompjuterik për shfrytëzuesit, sistemi operativ ofron një pamje të përgjithshme dhe logjike për ruajtjen e informacionit. Sistemi operativ fshehë karakteristikat fizike të pjesës për ruajtje për të definuar një njësi të ruajtjes, **fajllin**. Sistemi operativ i pasqyron fajllat në medime fizike të ruajtjes dhe u qaset këtyre fajllave përmes pajisjeve për ruajtje.

Menaxhimi i fajllave është njëra nga pjesët më të dukshme të një sistemi operativ. Kompjuterët mund të ruajnë informacion në disa medime të ndryshme fizike. Memoriet me kapacitet më të madh (memoriet sekondare) janë të zakonshme, por mund të përdoren edhe memorie tjera që kanë madhësi edhe më të madhe (memoriet e jashtme). Secili prej këtyre medimeve kanë karakteristikat dhe organizimin e tyre fizik. Shumica prej tyre kontrollohen nga një modul (kontrollues), si rasti i kontrolluesit të diskut, që ka gjithashtu karakteristikat e tij të veçanta. Këto karakteristika përfshijnë shpejtësinë e qasjes, madhësinë, shpejtësinë e bartjes së të dhënave dhe metodën e qasjes (sekuenciale ose të rastësishme).

Fajlli është një grumbull i informatave të lidhura në mes vete të përcaktura nga krijuesi i fajllit. Zakonisht, fajllat paraqesin programe (në formë të kodit burimor dhe në formë të kodit të kompajluar) dhe të dhëna. Fajllat e të dhënave mund të jenë numerik, shkronjorë, shkronjorë dhe numerik si dhe binarë. Fajllat mund të jenë të thjeshtë (për shembull, fajllat tekstual) ose mund të jenë të formatuar (për shembull, me fusha të madhësisë fikse si një fajll mp3 i muzikës). Sigurisht, koncepti i fajllit është shumë më i gjerë.

Sistemi operativ e realizon konceptin e fajllit duke i menaxhuar pajisjet ruajtëse dhe kontrolluesit e tyre. Fajllat organizohen në formë të direktorieve për t'i bërë më të lehtë të përdoren. Në fund, në rastet kur shumë shfrytëzues kanë qasje në fajlla, mund të jetë e dëshirueshme të kontrollohet se cili shfrytëzues mund të ketë qasje në fajll dhe si mund t'i qaset fajllit (për shembull, lexim, shkrim, zgjerim).

Sistemi operativ është përgjegjës për këto aktivitete në lidhje me menaxhimin e fajllave:

- Krijimi dhe fshirja e fajllave
- Krijimi dhe fshirja e direktorieve për organizimin e fajllave
- Përkrahja e komandave elementare për punë me fajlla dhe direktorieve
- Pasqyrimi (vendosja) i fajllave në memoriet me kapacitet të madh
- Ruajtja e një kopjeje të fajllave në memorie që i ruajnë të dhënat edhe pas ndërprerjes së furnizimit me elektricitet

1.5.4 Menaxhimi i memorieve me kapacitet të madh

Sistemi kompjuterik duhet patjetër të ofrojë memorie me kapacitet të madh në mbështetje të memories DRAM. Shumica e sistemeve kompjuterike përdorin disqe HDD dhe pajisje NVM si memorie ku ruajnë programet dhe të dhënat. Shumica e programeve – përfshirë kompajlerët, ueb shfletuesit, programet për përpunim të teksteve dhe lojërat – ruhen në këto pajisje deri sa të vendosen në memorie. Programet pastaj përdorin pajisjet ruajtëse si burim dhe destinacion për përpunimin e të dhënave. Kështu, menaxhimi i duhur i pajisjeve ruajtëse me madhësi të madhe është i një rëndësie qendrore për një sistem kompjuterik.

Sistemi operativ është përgjegjës për këto aktivitete në lidhje me menaxhimin e pajisjeve me madhësi të madhe:

- Vendosja dhe largimi në sistem
- Menaxhimi i hapësirës së lirë
- Planifikimi i përdorimit të diskut
- Ndarja e diskut në pjesë
- Mbrojtja

Meqë memoria sekondare përdoret shpesh dhe gjerësisht, duhet patjetër të përdoret në mënyrë efikase. Shpejtësia e punës së sistemit kompjuterik mund të varet nga shpejtësitë e nënsistemit të memorieve me madhësi të madhe dhe nga algoritmet që punojnë me nënsistemin.

Në sistem kompjuterik përdoren edhe memorie tjera më të ngadalshme dhe më të lira (ndonjëherë me madhësi më të madhe) se sa memoriët sekondare. Ruajtja e një kopjeje e të dhënave që janë në disk, ruajtja e të dhënave që përdoren më rrallë dhe ruajtja për një kohë (arkivimi) më të gjatë janë disa nga shembujt. Njësitë me shirita magnetik, CD-të, DVD-të, njësitë Blu-ray dhe pllakat janë memorie të zakoshme të rendit të tretë.

Memoria e rendit të tretë (memoria e jashtme) nuk është qenësore për performancën e sistemit kompjuterik, por duhet patjetër të menaxhohet. Disa sisteme operative merren edhe me këtë punë, e disa tjera punën e menaxhimit të memorieve të rendit të tretë ua lënë programeve aplikative. Disa prej funksioneve që mund të ofrojnë sistemet operative përfshijnë vendosjen dhe largimin prej sistemit të pajisjeve ruajtëse, dhënia dhe marrja e hapësirës së memories nga pajisjet dhe dhënia e saj për përdorim nga proceset dhe bartja e të dhënave nga memoriët sekondare në memoriët e rendit të tretë.

1.5.5 Menaxhimi i keshit

Keshimi është një parim i rëndësishëm në sistemet kompjuterike. Informacioni gjatë punës së kompjuterit ruhet në ndonjërin prej memorieve (në memoriën DRAM). Gjatë përdorimit, kopjohet në një sistem më të shpejtë të ruajtjes – në kesh – përkohësisht. Kur na duhet një informacion i caktuar, së pari kontrollohet se a është në kesh. Nëse është aty, informacioni merret prej keshit. Nëse nuk është aty, informacioni merret prej një memorie tjetër (DRAM) duke e ruajtur një kopje në kesh me supozimin se do të na duhet përsëri.

Përveç kësaj, regjistrat e CPU-së përdoren si kesh i shpejtësisë së madhe për memoriën DRAM. Programuesi (ose kompajleri) realizon algoritme për dhënie dhe zëvendësim të regjistrave për të vendosur se cilin informacion duhet mbajtur në regjistra dhe cilin në memoriën DRAM.

Disa keshe tjera realizohen plotësisht në harduer. Për shembull, shumica e sistemeve kompjuterike kanë një kesh për instruksione për ruajtjen e instruksioneve që pritet të ekzekutohen në radhë. Pa këtë kesh, CPU-ja do të duhej të priste disa cikle deri sa një instruksion të lexohej prej memories DRAM. Për arsyet e njëjta, shumica e sistemeve kanë një ose më shumë keshe të shpejta për të dhëna në hierarkinë e memories.

Meqë keshet kanë madhësi të kufizuar, **menaxhimi i keshit** është një problem i rëndësishëm i dizajnit. Përzgjedhja me kujdes e madhësisë së keshit dhe algoritmit për zëvendësim mund të rezultojë në rritje të performancës.

Bartja e informacionit në mes të niveleve të hierarkisë së memories mund të jetë eksplicite ose implicite, varësisht nga dizajni i harduerit dhe softueri kontrollues i sistemit operativ. Për shembull, bartja e të dhënave prej keshit në regjistrat e CPU-së zakonisht është një funksion i harduerit, pa përfshirjen e sistemit operativ. Për dallim, bartja e të dhënave nga disku në memorien DRAM zakonisht kontrollonhet nga sistemi operativ.

Në një strukturë hierarkike të memorieve, të dhënat e njëjta mund të ndodhen në nivele të ndryshme të hierarkisë. Për shembull, le të supozojmë se një integjer (numër i plotë) A që duhet të rritet (inkrementohet) për 1 ndodhet në fajllin B dhe fajlli B ndodhet në hard disk. Operacioni i inkrementimit së pari lëshon një komandë të hyrje/daljes për të kopjuar bllokun në të cilin ndodhet numri A prej diskut në memorien DRAM. Ky operacion shoqërohet me operacionin e vendosjes së një kopjeje të numrit A në kesh dhe pastaj në një regjistër të CPU-së. Në këtë mënyrë, kopja e numrit A ndodhet në disa vende: në hard disk, në memorien DRAM, në kesh dhe në njërin prej regjistrave të CPU-së. Kur kopja e numrit A që ndodhet në regjistër rritet për 1, vlera e numrit A dallon në memoriet e ndryshme. Vlera e numrit A bëhet e njëjtë vetëm pasi vlera e re e numrit A shkruhet prej regjistrat në hard disk.

Në një ambient kompjuterik ku vetëm një proces ekzekutohet në një moment të caktuar kohor, kjo zgjidhje nuk paraqet ndonjë vështirësi, meqë qasja në vlerën e plotë A bëhet çdo herë në kopjen që është në nivelin më të lartë të hierarkisë. Megjithatë, në një ambient multitasking, ku CPU-ja kalon prej njërit proces në tjetrin në mes të proceseve të ndryshme, kujdes i madh duhet të tregohet nëse disa procese duan t'i qasen vlerës A, atëherë secili nga këto procese do të marrë vlerën më të fundit të numrit A.

Gjendja bëhet edhe më e vështirë në një ambient me shumë procesorë, ku përveç mirëmbajtjes së regjistrave, secili nga CPU-të ka keshin e vet. Në një ambient si ky, një kopje e vlerës A mund të jetë në disa keshe në të njëjtën kohë. Meqë disa CPU të gjithë mund të ekzekutojnë paralelisht, duhet patjetër të sigurohemi që një ndryshim në vlerën A në njërin prej kesheve bartet menjëherë në të gjitha keshet tjera ku ndodhet numri A. Kjo gjendje quhet **koherenca e keshit** dhe zakonisht është një çështje harduerike (që trajtohet përfundimisht nivelit të sistemit operativ).

Në një ambient të shpërndarë (distributed), gjendja bëhet edhe më e vështirë. Në këtë ambient, disa kopje të të njëjtë fajll mund të mbahen në kompjuterë të ndryshëm. Meqë disa kopje mund të qasen dhe ndryshohen në mënyrë konkurente, disa sisteme të shpërndara sigurohen që kur njëra kopje ndryshohet në një vend, të gjitha kopjet tjera marrin vlerën e fundit me ndryshim sa më herët që është e mundur.

1.5.6 Menaxhimi i sistemit të hyrje/daljes

Njëri nga qëllimet e një sistemi operativ është të fshehë detajet e pajisjeve të caktuara harduerike nga shfrytëzuesi. Për shembull, në sistemin operativ UNIX, detajet e pajisjeve të hyrje/daljes

fshehen nga pjesa tjetër e sistemit operativ përmes **nënsistemit të hyrje/daljes**. Nënsistemi i hyrje/daljes përbëhet nga disa pjesë:

- Pjesa për menaxhim të memories që përfshinë baferimin, keshimin dhe spooling
- Një interfejs i përgjithshëm për drajverin e pajisjes
- Drajverë për pajisje harduerike të caktuara

Vetëm drajveri i pajisjes i din detajet e pajisjes së caktuar të cilën e menaxhon.

Për ndërtimin eficient të nënsistemit të hyrje/daljes përdoren procedurat (funksionet) për ndërprerje (interrupt handlers) dhe drajverët e pajisjeve.

1.6 Siguria dhe mbrojtja

Nëse një sistem kompjuterik ka shumë shfrytëzues dhe lejon ekzekutimin konkurent të shumë proceseve, qasja në të dhëna duhet patjetër të rregullohet. Për atë qëllim, mekanizmat sigurohen që fajllat, segmentet e memories, CPU dhe resurset tjera mund të përdoren vetëm nga ato procese që kanë autorizimin e duhur nga sistemi operativ. Për shembull, hardueri për adresim të memories sigurohet që një proces mund të ekzekutohet vetëm brenda hapësirës së vet të adresave. Tajmeri siguron që asnjë proces nuk mund të marrë kontrollin mbi CPU nëse dikur nuk e kthen prapa kontrollin. Regjistrat për kontroll të pajisjes nuk janë të qasshëm nga shfrytëzuesi, ashtu që integriteti i pajisjeve të ndryshme harduerike mbrohet.

Mbrojtja është çdo mekanizëm për kontrollimin e qasjes së proceseve ose të shfrytëzuesve në resurse të përcaktuara nga sistemi kompjuterik. Ky mekanizëm duhet patjetër të ofrojë një mundësi të vendosjes dhe zbatimit të kontrolleve.

Mbrojtja mund të përmirësojë besueshmërinë duke gjetur gabime të fshehura në interfejset në mes të nënsistemeve. Gjetja e hershme e gabimeve në interfejs (pikat e kontaktit) shpesh mund të parandalojë infektimin e një nënsistemi të shëndoshë nga një nënsistem tjetër i cili është duke dështuar. Për më tepër, një resurs i pambrojtur nuk mund të mbrohet nga përdorimi (ose keqpërdorimi) nga një shfrytëzues i paautorizuar ose jo kompetent. Një sistem i orientuar kah mbrojtja ofron një mundësi për të bërë dallimin në mes të përdorimit të autorizuar dhe të paautorizuar.

Një sistem mund të ketë mbrojtjen e duhur, por mund të jetë i dështueshëm dhe mund të lejojë qasje jo të duhur. Le të marrim një shfrytëzues informacioni autentikues (mënyra se si ajo e identifikon veten në sistem) i të cilës është vjedhur. Të dhënat e saj do të mund të kopjoheshin ose fshiheshin, edhe pse mbrojtja e fajllit dhe memories janë duke punuar. Është punë e **sigurisë** mbrojtja e një sistemi nga sulmet e jashtme dhe të brendshme. Sulmet e tilla mund të jenë të ndryshme dhe përfshijnë viruset dhe worms, sulmet për ndalim të shërbimit (të cilat përdorin të gjitha resurset e sistemit dhe mbajnë jashtë sistemit shfrytëzuesit legjitim), vjedhja e identitetit dhe vjedhja e shërbimit (përdorim i paautorizuar i një sistemi). Parandalimi i disa prej këtyre sulmeve konsiderohet si funksion i sistemit operativ në disa sisteme, përderisa disa sisteme tjera ua lënë atë softuerëve tjerë. Për arsye të rritjes alarmuese të incidenteve të sigurisë, karakteristikat e sigurisë në sistem operativ janë një fushë me rritje të shpejtë e kërkimit dhe implementimit.

Mbrojtja dhe siguria kërkojnë që sistemi të jetë në gjendje të dallojë shfrytëzuesit e tij. Shumica e sistemeve operative mirëmbajnë një listë të emrave të shfrytëzuesve dhe **identifikuesin e shfrytëzuesit (user ID)**. Në zhargonin e Windows-it, kjo është një **ID e siguri** (**security ID – SID**). Këto ID numerike janë unike, një për secilin shfrytëzues. Kur shfrytëzuesi logohet në sistem, faza e autentikimit përcakton ID-në e duhur të shfrytëzuesit për shfrytëzuesin. Ajo ID e shfrytëzuesit lidhet me të gjitha proceset dhe thredat e shfrytëzuesit. Kur një ID duhet të jetë e lexueshme nga një shfrytëzues, ID përkthehet në emër të shfrytëzuesit përmes listës së emrave të shfrytëzuesve.

Në rrethana të caktuara, ne duam të dallojmë grupet e shfrytëzuesve në vend se të secilit shfrytëzues individual. Për shembull, shfrytëzuesi pronar i një fajlli në një sistem UNIX mund të lejohet të përdorë të gjitha operacionet mbi atë fajll, përderisa një grup i përzgjedhur i shfrytëzuesve mund të lejohet vetëm të lexojë atë fajll. Për të arritur këtë, duhet të përcaktojmë një emër grupi dhe bashkësinë e shfrytëzuesve që i takojnë atij grupi. Funksionaliteti i grupit mund të realizohet si një listë e emrave të grupeve në nivel të sistemit dhe **identifikuesit të grupit**. Një shfrytëzues mund të jetë në një ose më shumë grupe, varësisht prej vendimeve që janë marrë gjatë dizajnit të sistemit operativ. ID-të e grupit të shfrytëzuesit përfshihen edhe në secilin proces dhe thred shoqërues.

Gjatë punës së një sistemi kompjuterik, ID e shfrytëzuesit dhe ID e grupit janë të mjaftueshme për një shfrytëzues. Megjithatë, një shfrytëzues mund të ketë nevojë të **zgjerojë privilegjet** për të marrë të drejta shtesë për ndonjë aktivitet. Sistemet operative ofrojnë metoda të ndryshme të zgjerimit të privilegjeve. Në UNIX, për shembull, atributi *setuid* në një program bën që ai program të ekzekutohet me ID-në e shfrytëzuesit pronar të fajllit, në vend se me ID-në e shfrytëzuesit aktual. Procesi ekzekutohet me këtë **UID efektive** derisa të heqë dorë prej privilegjeve shtesë ose të përfundojë ekzekutimin.

1.7 Virtualizimi

Virtualizimi është një teknologji që mundëson të fshehim (abstrahojmë) harduerin e një kompjuteri (CPU-në, memorien, disqet, kartelat e rrjetës dhe pjesë tjera) në disa ambiente të ndryshme të ekzekutimit, duke krijuar në atë mënyrë iluzionin që secili ambient punon në kompjuterin e vet. Këto ambiente mund të trajtohen si sisteme operative të ndryshme (për shembull, Windows dhe UNIX) që mund të ekzekutohen në të njëjtën kohë dhe mund të bashkëpunojnë me njëri-tjetrin. Një shfrytëzues i një **makine virtuale (virtual machine)** mund të zgjedhë në mes sistemeve të ndryshme operative në mënyrën e njëjtë sikur një shfrytëzues që mund të zgjedhë në mes të proceseve të ndryshme që ekzekutohen në mënyrë konkurente në një sistem operativ.

Virtualizimi mundëson që sistemet operative të ekzekutohen si aplikacione brenda sistemeve operative tjera. Në shikim të parë, mund të duket se ka pak nevojë për një funksion të tillë. Industria e virtualizimit është e madhe dhe në rritje, gjë që tregon përdorimin dhe rëndësinë që ka.

Në terma të përgjithshëm, softueri për virtualizim është njëri nga anëtarët e një klase që përfshinë edhe emulimin. **Emulimi**, që përshinë simulimin e harduerit të një kompjuteri përmes softuerit, zakonisht përdoret kur CPU burimor është tjetër prej CPU-së target. Për shembull, kur Apple kaloi nga procesorët Power PC të IBM në procesorët x86 të Intel-it për kompjuterët e vet desktop dhe laptop, Apple përfshiu një mundësi emulimi të quajtur “Rosetta”, që lejoi programet e kompajluara për procesorët e IBM-it të ekzekutohen në procesorët e Intel-it. Koncepti i njëjtë mund të përdoret që krejt sistemi operativ i shkruar për një platformë harduerike të ekzekutohet në një platformë tjetër. Megjithatë, emulimi ka çmimin e vet të lartë. Çdo instruksion i nivelit të makinës (në gjuhën e makinës) që ekzekutohet në CPU-në për të cilin është shkruar programi duhet patjetër të përkthehet në funksionin ekuivalent për CPU-në target, gjë që zakonisht jep shumë instruksione target. Nëse CPU-të burimor dhe target (destinacion) kanë nivele të ngjashme të performancës, kodi i emuluar mund të ekzekutohet më ngadalë se sa kodi i shkruar për procesorin burimor.

Përmes virtualizimit, për dallim, një sistem operativ që kompajlohet për një CPU burimor ekzekutohet brenda një sistemi operativ poashtu burimor për atë CPU. Virtualizimi së pari u realizua në kompjuterët mainframe të IBM si një metodë që shumë shfrytëzues të ekzekutojnë punët në mënyrë konkurente. Ekzekutimi i shumë makinave virtuale lejoi (dhe lejon ende) shumë shfrytëzues të ekzekutojnë punë në një sistem të dizajnuar për vetëm një shfrytëzues. Më vonë, si përgjigje ndaj problemeve të ekzekutimit të shumë kopjeve të Microsoft Windows në procesorët Intel x86, VMware krijoi një teknologji të re të virtualizimit si një aplikacion që u ekzekutua në Windows. Ai aplikacion ekzekutoi një ose më shumë kopje **mysafire (guest)** të Windows-it ose të sistemeve operative tjera për x86, secili prej tyre duke ekzekutuar aplikacionet e tyre. (Shikoni në Figura 1.6).

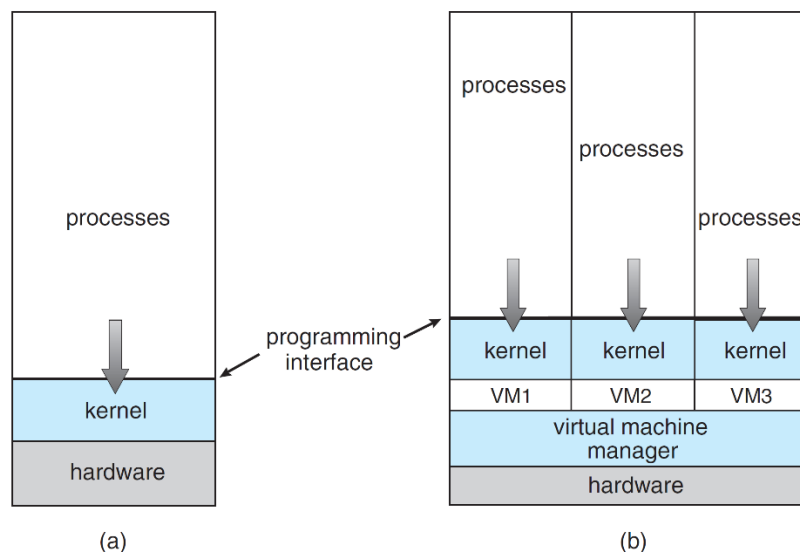


Figura 1.6 Një kompjuter duke ekzekutuar (a) vetëm një sistem operativ dhe (b) tre makina virtuale

Windows-i ishte sistemi operativ **nikoqir (host)** dhe aplikacioni VMware ishte **menaxhuesi i makinës virtuale (VMM)**. VMM ekzekuton sistemet operative mysafire, menaxhon përdorimin e resurseve dhe mbron secilin mysafir nga tjerët.

Edhe pse sistemet operative moderne janë plotësisht në gjendje të ekzekutojnë besueshëm shumë aplikacione, përdorimi i virtualizimit vazhdon të rritet. Në laptopë dhe desktopë, një VMM lejon shfrytëzuesin të instalojë shumë sisteme operative për qëllime studimi ose për ekzekutim të aplikacioneve të shkruara për sisteme operative tjera nga sistemi operativ nikoqir. Për shembull, një laptop Apple që ekzekuton sistemin operativ macOS në një CPU x86 mund të ekzekutojë sistemin operativ Windows 10 për të mundësuar ekzekutimin e aplikacioneve të Windows-it. Kompanitë që zhvillojnë softuer për shumë sisteme operative mund të përdorin virtualizimin për të ekzekutuar të gjitha ato sisteme operative në një server të vetëm fizik për zhvillim, testim dhe gjetje e përmirësim të gabimeve (debugging). Në qendrat e të dhënave, virtualizimi është bërë një metodë e zakonshme për ekzekutim dhe menaxhim të ambienteve kompjuterike. VMM-të si VMware ESX dhe Citrix XenServer nuk ekzekutohen më në sisteme operative, por **janë** sisteme operative nikoqire, duke ofruar shërbime dhe menaxhim të resurseve për proceset e makinave virtuale.

Referenca

Materiali është marrë nga kapitulli 1 (faqe 3) i librit **Operating System Concepts** i autorëve **Abraham Silberschatz, Peter Baer Galvin dhe Greg Gagne**.