VIRTUALNI POMNILNIK: Je na vsakem sistemu, vendar zastarelo in obstajajo samo zaradi združljivosti z starimi programi. Izhaja iz časov, ko je bilo na voljo zelo malo delavnega pom.
OVERLAY:program je samo razbil prog. na več enot, kjer vsaka ni presegala pom. ki je bil na voljo, med enotami ni bilo povezav. Se pravi izvede se prvi del, shrani rezultate, naprej se izvaja drugi del. To je avtomatizacija tega postopka: program ne potrebuje več sam deli programe, to naredi sistem (prog. se razdeli na page). Ustvari polje navideznega naslova, preslikovalnik jih preslika v fizični naslove, kjer se dejansko nahajajo podatki.
Preslikovalnik: Če vsega potrebnega ne moreš spraviti na RAM , mora ostati na disku.CPE pa ne more dostopati do diska, zato se3 rabi preslikovalnik. Ta vodi evidenco na vseh straneh(kjeso, katere so).Lahko tudi nastaviš velikost strani,a ni priporočljivo.Preslikovalnik omogoča tudi asociacijo pom. naslovo v trenutku izvajanja.
SEKUNDARNI POMNILNIK: najpogostejši je disk. Namenjen je trajno shranjevanju pod. Vsaka plošča ima 2 glavi, ki bereta podatke.Na disku so podatki v krožnicah. Vse glave se premikajo hkrati.Vaka bere eno sled, skupaj berejo cilinder.Sled se razdeljena na sektorje. Sektor ima gavo, rep in vmes podatki. ***CRC*** pove če je prišlo pri pisanju/branju do napake. Napake pri zapisih na sektorjih se lahko zgodijo če nap. bi rep prepisal glavo naslednjega sektorja in bi tako še huje pokvaril stvari. ***GLAVA:*** se nahaja zelo blizu površine diska, vsake nečistoče na disku povzroči napake.Ko glava naleti nanjo se dvigne in ko se spušča lahko poškoduje disk. ***Hitrost vrtenja diska:*** od nje je odvisen dostopni čas.Ob večji hitrosti je večje pregrevanjezato jeta omejena. Sledi so razločno dolge(daljše so na zunanji strani).
SEKTORJI: -so enako veliki a različno dolgi. Biti na zunanji sledi so daljši.Vsi sektorji hranijo enako količino podatkov.Dostop do vseh sektorjev je enako hiter.
ZAPISOVANJE DATOTEK NA DISK:
*ZAPOREDNO ALOCIRANJE:*Za zapis datoteke se rezervirajo zaporedni bloki na disku.Je enostavno, za dostop do datoteke potrebujemo samo začetek in dolžino datoteke.
PROBLEMI: -Širjenje datotek: Kar je za datoteko, jetreba prestaviti da naredimo prostor za podaljšek. Defragmentacija- zapolnitev lukenj na disku.
POVEZANO ALOCINIRANJE: Podatkovni bloki dat. So razpršeni po disku. V datotečnem imeniku imamo samo kazalec na prvi blok. Vsak blok vsebuje kazalec na naslednji blok. Končani blok vsebuje zaključek. V teoriji lahko vedno datoteke vedno širimo, ni potrebna defragmentacija. V praksi je potrebna defragmentacija, zaradi optimizacije dostopa. Bloki za isto datoteko naj bodo čim bližje skupaj zaradi hitrejšega dostopa do celotne dat.
SLABOSTI: če želimo prebrati le del datoteke, moramo vseeno preiskati celo dat., saj poznamo le začetni blok.
INDEKSIRANO ALOCIRANJE: Rezervira se le indeksni blok, ki vsebuje kazalec na vse bloke datoteke. V datotečnem imeniku imamo kazalec na indeksni blok. Dostop je hiter , v dveh korakih lahko prideš do katerega koli bloka. Napaka v enem bloku ne pokvari celotne dat. (razen če je to indeksni blok).
DATOTEČNI SISTEMI: Zagotavlja mehanizme za datoteke, pri čemer omogoča lažjo organizacijo iskanja. Datoteka: abstraktni podatkovni tip, ne opredeljuje tipa podatkov, shranjenih v dat.
IME datoteke*:*Zunanje ime, s katerim operira uporabnik. Ga potrebujemo, da lahko dat. najdemo.Lahko že obstaja a je ne moremo najti v množici datotek. Proces ne uporablja tega imena (uporablja proces ID). ***TCB*** shrani zunanja imena programov, da jih lahko kasneje poišče če jih potrebuje. Omejitev je 255 znakov. Windows niso case sensitive, Linux so.
TIP-končnica: opisuje podatke, ki se nahajajo v datoteki. V Windowsih so programivezani na končnico dat.(ta določa kateri program jo bo odprl). PROBLEM: Vsakdo lahko na hitro pogleda kaj je v dat., virusi se enostavno zaženejo(jpg. Exe).Pri Linoxu so tipi opcijski. Zaradi komercializacije tudi takoj podoben kot priwindowsih.Varneje je z virusi in uporabnik ima manj pravic.

LOKACIJA: Kazalec na napravo in mesto znotraj naprave kjer se nahaja datoteka.Je sistemski podatek, ki pove kje naj sistem išče podatke iz datoteke.
VELIKOST: Služi za informacijo uporabnikom.Služi sistemu pri kopiranju datotrkr v primarni pomn., zato je treba v primarni pomn. rezervirati prostor zanje. Če je prevelik se del pusti v navideznem pomn.
ZAŠČITA: FAT 32- samo za branje, skrita, arhivska. NTSF- read, write, execute. Linux ima boljšo zaščito:določeno za lastnika datoteke, skupine uporabnika posebej. Zaščita omogoča stabilnost sistema(pri linoxu kot uporabnik nimaš pravice spreminjati sistemskih dat.)
ČAS:Čas tvorbe, čas spremembe, čas zadnje uporabe je uporaben za porabnika. Glede na ta atribut se lahko shranjujejo datoteke kot verzije ki se po njem lahko išče.
LASNIŠTVO: -kdo je lastnik datoteke.Lastnik lahko z datoteko dela kar koli želi.Pri linuxih je lastnik za sistemske dat. ROOT. Pri Windowsih pa je SYSTEM.
DIREKTORIJI: združujejo skupaj sorodne datoteke.Zaseda posamezno logično podatkovno enoto.*LOKACIJA DATOTEK:* shranjena v simbolični tabeli.Iskanje, kreiranje,brisanje, preimenovanje in prehodi med posameznimi direktoriji morajo biti enostavni.
ORGANIZACIJA DIREKTORIJEV: WINDOWS- so bližnjice se pravi povezave na isto datoteko na različnih mestih. Hkrati lahko ista datoteka obstaja v večih direktorijih(2 povezavi kažeta na isto datoteko). LINUX- na datoteki obstaja števec povezav na datoteko, ki se poveča ob dodani povezavi.
INODE: Vsak inode obsega seznam diskovnih blokov, ki pripadajo posamezni dato.(velja za manjše dat.).Sam inode ne vsebuje imen datotek in direktorijev, ti so shranjeni v imeniški strokturi (ime datoteke + inode,kjer se ta nahaja). Zato lahko ima dat. v linuxih več imen, na večih mestih.Novo ime se samo doda v imeniško strokturo.
PORAZDELJENI DATOTEČNI SISTEMI: Odjemalci, pomnilnik, strežniki so porazdeljeni po lokalnem rač. omrežju, med seboj se lahko razlikujejo. Ko se uporabnik usede za rač. mora imeti občutek da je vse na njegovi napravi. Namen je za poslovno okolje, timsko delo…
LASNOSTI: Transparentnost-za odjemalca ni važno, kje datoteka dejansko je(fizično) Neodvisnost od lokacije:datoteka ostane v istem direktoriju če jo prestavimo iz enega diska do drugega.Mobilnost uporabnikov- dostop do dat. iz različnih rač. Zmogljivost: hitrost se zmanjša za kolikor potrebuje mreža da dostavi dat. Toleranca izpadov: sistem je ranljiv na večih mestih (izpadi strežnikov, mreže..), če pade odjemalčeva mreža, se pač prestavi na drugo(podatki niso shranjeni lokalno).
FCFS-First come first served: Zahtevki se obdelajo v tistem vrstnem redu v katerem se pojavijo. Je poštena a ne preveč učinkovita.
SSTF-Shortest seek time first:Najprej se obdela tisti, ki je najbližji.Problem: oddaljene inf. Problem stradanja: Zahtevki po oddaljenih inf. ne pridejo na vrsto.
SCAN- Glava potuje po vseh sledeh in sproti obdeluje zahtevke, na katere naleti. Problem: še vedno deli,kjer je veliko zahtevkov in traja dolgo, da se to obdela.
N-SCAN-next scan: Scan ki gre čez zahtevke in ne upošteva novonastalih. Glava še vedno potuje v eni smeri.
C-SCAN-circular scan: Najbolje uravnoteži pristope.Podobno kot scan vendar ko pride do konca ne gre po isti poti nazaj, ampak skoči na začetek. Nap.: scan: ms-lj-kp-lj-ms; c-scan: ms-lj-kp//ms-lj-kp.
VHODNO-IHODNI SISTEM: Device controller- pretvarjenje podatkov iz oblike, ki jjo uporablja operacijski sistem v obliko, primerno za predstavitev na 10 napravah. In obratno. Controller: v/i enote-krmilniki, ide kontroler; Device: v/i- miš, monitor, printer, cd enote.
Za V/I operacije mora CPU v registre krmilnika vpisati določeno vrednost. Krmilnik operacijio izvrši in sporoči, kako je operacija bila izvedena (uspešno/neuspešno).
SINHRONI način dela: preden nadaljujemo z delom, počakamo da se operacija izvrši in šele po potrditvi nadaljujemo.

ASINHRONI način dela: ne čakamo na izvrševanje operacije, takoj nadaljujemo z delom.
TABELA: Sistem potrebuje prosti IRQ- vsaka naprava rabi svojega.
KRMILNIŠKI OBSEG: OS določi, preko katerega dela pomnilnika bo komunicirala z krmilnikom.PROBLEM: dve napravi hkrati želita dostop krmilnika, OS mora rešiti, katera naprava dobi dostop prva.Če napišemo napačno vrednost v krmilnik, lahko sesujemo celotni sistem- Zato OS ne dovoli dostopa do hardwera.
OS mora zagotoviti čim večjo neodvisnost krmilnikov in V/I naprav.Deli-splošen vmesnik do gonilnika;-vmesno polje za pomnenje podatkov; posamezni gonilniki V/I naprav. Gonilniki niso del operacijskegasistema.Se le vsedejo na splošen vmesnik in preko njdega povežejo z OS.Zato je sistem odprt-Lahko dodajamo nove gonilnike in s tem nove naprave.
Primer branja enega znaka:kaj se zgodi, ko prog. pride do ukaza CIN:
1.)Program naroči OSmu, naj mu vrne tipke, ki je bila pritisnjena na tipkovnici.Proces gre v stanje blokiran.(C1)
2.)Jedro prejme klic, proces postavi na blocked.Reče gonilniku, naj mu dostavi tipko, ki je bila pritisnjena.Gonilnik ima Buffer, v katerem je lahko že shranjen znak.Če je, se ta posreduje naprej.(C2)
3.)Pritisnemo tipko-koda tipke se shrani v tipko.Kotroler generira prekinitveno zahtevo,ki štarta(16).Prekinitvena servisna rutina ki jo štarta CPE, poskrbi za prenos kode iz V/I naprave v gonilnikov buffer(17-18).Zatem obvestimo device driver, da je sedaj tipka v bufferju (19), ta zatem obvesti CPE,da je dobil tipko(C12) in dostavi kodo tipke procesu(C11).
4.)Jedro zatem zbudi proces in ga vrne v stanje izvajanja.
(Pri sliki): C1 je različen glede na to, kaj iščemo(niz,število). Če beremo niz, se bodo znaki zbirali dokler ne pride še Enter, šele potem se proces zbudi.
ZAŠČITA IN VARNOST OS: Možnost zaščite, realizirana preko matrike dostopnosti.Računalniški sistem je ponazorjen z množico objektov.Objekti so ali operativni ali programski.V ta sistem uvedemo domeno zaščite.Proces ki se kreira, se uvrsti v določeno domeno in s tem se določi, kaj lahko dela in s katerim lahko dela.Prede3n se procesu dovoli akcija, se pogleda v matriko dostopnosti.

	01	02	03	04
D1	R			
D2				
D3		R	izv	W/R
D4	W/R			

	D1	D2	D3	D4
D1		preklop		
D2			prek	prek
D3				
D4	preklop			

Tudi ta tabela je objekt, ki je zaščiten samo sistem ima dostop do nje.

Ta zaščita je interne narave (izhaja iz OS) in ščiti OS pred procesi.Varnost OS vključuje tudi okolico OS-a, tudi fizično zaščito, varnost pod. Baz(omejevanje tipa podtkov). Operacijaz napačnimi podatki bi lahko povzročile nestabilnost sistema.
NALOGE IN POSTOPKI SISTEMSKEA ADMINISTRATORJA:-dodajanje in brisanje uporabnikov,dodajanje novih strežnikov na mrežo, arhiviranje pod.,nameščanje popravkov obstoječega softwarea, nameščanje novega softwarea, nadziranja učinkovitosti sistema, nadzor tiskanja, iskanje varnostnih lukenj v sistemu,pisanje skript za avtomotizacijo dela, zagotavljanje delovnega sistema, podpora uporabnikom, iskanje sistemskih napak, vzdrževanje strojne opreme, vzdrževanje prog. opreme, varnostna politika (sistem uporabniških imen in gesl, identifikacija delovnih postaj, protivirusni programi, požarni zid, varnostni popravki), vzdrževanje informacijskih servisov(strežniki za mail…)