

1. Kaj je OS?

OS je program, ki deluje kot posrednik med uporabnikom in strojno opremo. Pri tem skuša:

1. zagotoviti učinkovitost uporabe strojne opreme
2. ustvariti prijazno okolje

2. Kaj sestavlja računalniški sistem?

1. uporabniki
2. uporabniški programi
3. strojna oprema
4. operacijski sistem

3. Lastnosti OS

1. dodeljuje vire programom skladno s potrebami in zmožnostmi
2. rešuje konfliktna stanja
3. optimizira uporabo računalniških virov
4. nadzira delo nekaterih virov
5. nadzira izvajanje upor. programov
6. olajšuje delo uporabniku
7. povečuje učinkovitost računalniškega sistema

4. Rezidentni monitor

Uporablja se za avtomatizacijo delovanja poslov. Ob koncu vsakega posla kliče naslednjega. Ob zagonu se sam naloži v pomnilnik ter ostane tam celi čas. Za delovanje potrebuje kontrolne kartice, zato vsebuje tudi interpreter kartic, nalagalnik ter gonilnike za V/I naprave.

5. Prekrivanje operacij

Realizirano je bilo z ločenim V/I ali pa preko spoolinga. Ločeni V/I se je pojavil, ko se je pojavil magnetni trak, saj je trak bil vmesni člen med V/I in CPU. Tako so lahko hkrati potekale V/I in računske operacije. Problem je bil, ker je trak imel zaporedni dostop.

6. Razloži spooling!

Spooling se je pojavil z diskom, saj imajo ti neposredni dostop. Uporablja branje vnaprej – s tem omogoči, da je na disku več pripravljenih poslov hkrati. Omogoča, da se disk lahko uporabi kot velik vmesnik za začasno hranjenje vhodno/izhodnih podatkov dokler V/I naprava ni pripravljena in prekrivanje V/I operacij ter računskih operacij.

7. Multiprogramska paketna obdelava

Smisel je, da ko se posel konča ali ustavi, OS izbere novega. S tem se pojavijo zahteve po razvrščanju poslov, upravljanju s pomnilnikom, dodeljevanju procesorja (razvrščanju na procesorju) ter zaščiti.

8. Časovno dodeljevanje – večpravični OS

Multiprogramski sistem paketne obdelave, ki uporablja še timer. Hkrati lahko dela več uporabnikov, vsak s svojim poslom, toda vsak posel se sme neprekinjeno izvajati omejen čas. OS izbere posel, mu dodeli čas za izvajanje in ga pošlje, če mine dodeljeni čas, ali pa se posel ustavi (končal, čaka na človeka) prevzame nadzor OS, ki izbere drug posel, mu dodeli čas, in ga pošlje. Pri tem se pojavijo zahteve po upravljanju z zunanjim pomnilnikom, navideznim pomnilnikom, zahteve po datotečnem sistemu, po zaščiti, po dodeljevanju procesorja ter po sinhronizaciji in komunikaciji.

9. Krmilnik

Nadzoruje ustrezno napravo ali več naprav.

10. Bootstrap

Majhen, enostaven program, ki inicializira računalniški sistem (registre CPE, pomnilnike, krmilnike...), naloži jedro OS v pomnilnik in mu nato preda nadzor.

11. Vrste prekinitev

Strojna, sproži strojna oprema: Npr. naprava pošlje po vodilu prekinitveno zahtevo in z njo prekine tekoče delo na CPE, konec V/I operacije, neveljaven poseg v pomnilnik.

Programska, če jo sproži tekoči program: Npr. sistemski klic, s katerim tekoči program zahteva pomoč OS, past (trap), ker je CPE zaznala izjemno situacijo (deljenje z 0)

Ko se pojavi prekinitvena zahteva:

CPE prekine tekoče delo, izvede prekinitveno servisni program (v nadaljevanju PSP), s katerim se odzove na dogodek, nato nadaljuje prekinjeno delo.

12. Polling – programsko izpraševanje

Ko se pojavi prekinitiv, CPE prekine delo, zažene PSP ki ugotovi kdo je zahteval prekinitiv, zažene ustrezni PSP in nato nadaljuje z delom.

13. Vektorsko prekinjanje

Ko naprava zahteva prekinitiv, v CPE pošlje tudi indeks i, s pomočjo katerega nato CPE v prekinitvenem vektorju (tabeli v pomnilniku) poišče ustrezen PSP ter ga zažene.

14. Vhodno izhodni sistem

Sestavljen je iz V/I naprav, krmilnikov ter programov za V/I operacije. Krmilnik skrbi za podrobnosti pri prenosu podatkov – na njih se izvajajo V/I programi.

15. Sinhroni V/I

Proces, ki je zahteval V/I operacijo, čaka da se ta konča. Lahko izvede kako zanko, ukaz wait ali pa sam spremlja statusni register.

16. Asinhroni V/I

Proces, ki je zahteval V/I operacijo, nadaljuje svoje izvajanje. Če ne more nadaljevati dela, ker čaka na podatke, sam sebe blokira. Je primeren za počasne naprave.

17. DMA krnilnik

Se uporablja pri hitrih napravah za prenos med pomnilnikom in V/I napravami, brez posredovanja CPU.

CPU pošlje ukaz in podatke gonilniku, gonilnik podatke zapiše v registre DMA krmilnika in se blokira.

Ko DMA konča prenos pošlje prekinitveno zahtevo CPE.

18. Kje se uporablja tabela stanj naprav?

Uporablja se pri asinhronem V/I, kjer omogoča več vzporednih V/I operacij. Tabela vsebuje osnovne podatke o napravi in kazalec na tabelo V/I zahtev, kamor se vpisujejo vse zahteve za to napravo.

19. Elementi strojne zaščite

Sistemske način delovanja in uporabniški način delovanja

V/I naprave : V/I ukaze sme izvajati le OS, saj le ta pozna stanje V/I naprav (npr. preko tabele stanj naprav) V/I ukazi morajo zato biti privilegirani.

GI. pomnilnik : Nastavimo spodnjo in zgornjo mejo trenutno dovoljenih naslovov. Pred dostopom do pomnilnika se vsak naslov strojno preveri. Če je izven dovoljenega obsega, se program ujame v past (in nadzor prevzame OS).

CPE : CPE je treba zaščititi pred tem, da bi si jo upor. program prilastil – uvedemo timer.

20. Komponente OS

Upravljanje procesov - Ustvarjanje, uničenje, odložitev procesa, njegovo ponovno obuditev, usklajevanje med procesi, komunikacija med procesi, reševanje iz zastojev (mrtvih zank), in še kaj.

Upravljanje glavnega pomnilnika - Nadzor – dodeljevanje in odvzemanje

Upravljanje datotek - Ustvarjanje in brisanje, premeščanje, kopiranje, stik, sprehajanje po dir., preslikavanje med datoteko in resničnim zunanjim pomnilnikom, skladiščenje (backup), t.j. varnostno shranjevanje.

Upravljanje V/I sistema - Vsplošen vmesnik za komunikacijo z gonilniki, vsebovati gonilnike za posamezne V/I naprave, upravljati z vmesnimi pomnilniki (buffer), predpomnilniki (cache), spoolingom in še kaj.

Zaščita in varnost - Nadzorovanje dostopa procesov do posameznih virov in zaščita pred okolico.

Delo v omrežju

Interpretacija uporabnikovih ukazov

21. Sistemski klic

Z njegovo pomočjo programer zaprosi OS, da ta v njegovem imenu izvede neko (zanj prepovedano) privilegirano akcijo. Nekateri višji programski jeziki pa te klice (read, load, store, create, open, execute ...) vseeno omogočajo.

V grobem jih razberstimo v skupine : upravljanje procesov, upravljanje datotek, upravljanje naprav, vzdrževanje in dostavo informacij, komunikacijo

22. Zgradbe jedra?

1. Enostavna – monolitna - Jedro je veliko, opravlja veliko funkcij in je 'neurejeno'. ranljivost, nezanesljivost, težavna razširjenost. MSDOS, začetni UNIX
2. Razslojena - spodnji sloj je strojna oprema, vsak višji sloj pa temelji le na nižjih slojih. Skrivanje podrobnosti, prinaša red., manjša učinkovitost zaradi režije pri prenašanju. THE (6 slojev), Venus (7)
3. Mikrojedro - ohrani le bistvene komponente. Jedro je majhno, razširljivo, obvladljivo, prenosljivo. Celoten sistem pa enostavno razširljiv, zanesljiv, varen. Npr. če eden od servisov 'izpade', to ne prizadene celotnega sistema. UNIX, Apple MacOS, Windows NT.

23. Kaj je proces?

Proces je program v izvajanju. Je ustvarjen, teče in nato uničen. Za svoje delovanje potrebuje vire (CPE, pomnilnik, V/I, datoteke, čas, ...). Nekatere vire dobi ob nastanku, druge kasneje. Z okolico si lahko izmenjuje podatke. Hkrati lahko obstaja več procesov. Lahko so neodvisni, lahko pa se usklajujejo, komunicirajo. Običajno tekmujejo za omejene vire. Proces sam lahko ustvari nove procese. Program je pasivna entiteta - vsebina neke datoteke. Proces poleg programa zajem a tudi trenutno stanje nekaterih drugih entitet, ki se spreminjajo med izvajanjem programa.

24. Naštev stanja procesa v CPU!

Stanja procesa so:

ustvarjen (nov), pripravljen (čaka, da se mu dodeli procesna enota), teče (se izvaja), čaka (ustavljen), končan.

25. Kaj je PCB?

PCB je nadzorni blok procesa v OS-u. Vsebuje podatke o procesu, med njimi tudi kazalce za povezovanje z drugimi PCB v razne vrste. (stanje, prioriteta, podatki za obnovo, sklad, spremenljivke, podatki o pomnilniku, V/I in datoteke, ostalo)

26. V katerih vrstah se nahajajo procesi?

1. vrsta vseh procesov
2. vrsta pripravljenih procesov
3. vrsta procesov, ki čakajo na V/I napravo

27. Razloži razvrščanje – prekinitveno in neprekinitveno!

Ločimo tri vrste razvrščanja

1. na procesorju (kratkoročno – teče, pripravljen)
2. razvrščanje poslov (dolgoročno – disk, nov)
3. menjavanje (srednjeročno – disk, pripravljen in obratno).

Razvrščanje opravlja razvrščevalnik, ki je del OS.

Na enoprocesorskem sistemu v nekem trenutku teče kvečjemu en proces. Ostali procesi bodisi čakajo (na nek dogodek) ali pa so pripravljeni (čakajo na CPE). Med pripravljenimi se slednjič enega izbere in sedaj se njemu dodeli CPE, temu se reče **preklop**. Pravimo, da je procesor zamenjal okolje svojega delovanja. Tak preklop imenujemo tudi menjava okolja (context switch).

Katera dva programa OS sta udeležena pri menjavi okolja (preklopu)?

1. **razvrščevalnik** (scheduler), ki izbere enega od pripravljenih procesov,
2. **dodeljevalnik** (dispatcher), ki izbranemu procesu v resnici dodeli procesor, tj. izvede menjavo okolja.

V katerih okoliščinah je možno/nujno pognati razvrščevalnik?

1. tekoči proces je postal onemogočen - Proces ne more nadaljevati, zato bo šel med čakajoče in sprostil CPE.
2. tekoči proces je prekinila ura - Tekoči proces je rabil svojo časovno rezino.
3. čakajoči proces je dočkal izpolnitev pogojev za nadaljevanje - Čakajoči proces je postal spet pripravljen.
4. tekoči proces se je končal - CPE je prosta.
5. prispel je nov proces - Prispel nov proces, ki je pripravljen.

Katera dva pristopa ločimo?

Razvrščanje povzročita le 1 in 4. Brez prekinitev

Proces bo torej izgubil CPE le, če je bil onemogočen oz. se je končal, torej če ne more več teči. V nasprotnem pa bo nemoteno tekel. Zato to imenujemo **razvrščanje brez prekinjanja** (nonpreemptive sch.) (Pozor: Prekinitve, ki so posledica običajnih prekinitvenih zahtev, so seveda možne, saj ne sprožajo razvrščanja, temveč le začasno izvajanje prekinitvenih servisnih programov PSP - ti pa 'ne štejejo'. Zato bi bil ustrežnejši prevod razvrščanje brez odvzemanja, saj se procesu ne da odvzeti CPE, lahko pa se ga še vseeno prekine v standardnem smislu.)

Poleg 1 in 4 razvrščanje sproži še vsaj ena od 2,3,5. S prekinitvami

(Za odvzem sedaj zadošča tudi iztek dodeljenega časa ali pa sprememba v vrsti pripravljenih procesov.) V tem primeru proces lahko izgubi CPE, četudi bi še lahko tekel. Torej je lahko res prekinjen. To je **razvrščanje s prekinjanjem** (preemptive sch.)

28. Algoritmi za razvrščanje

1. **Prvi pride, prvi melje (FIFO)** – proces, ki je prej zahteval procesor ga prvi dobi
2. **Krožno razvrščanje (round robin)** – vrsta pripravljenih procesov je krožni seznam in proces, ki pride na vrsto dobi nek časovno rezino. Ko ta poteče ali pa se proces ustavi, se proces postavi na konec vrste
3. **Razvrščanje po prioriteti** – vsak proces ima prioriteto, najvišji dobi CPE
4. **Najkrajši nemoteni čas (SJF - Shortest Job First)** – CPE se dodeli procesu, ki bo najmanj časa nemoteno tekel. Napovedujemo tako, da predpostavimo, da bo naslednji nemoteni tek trajal podobno časa, kot trenutni
5. **Najkrajši nemoteni tek + prekinjanje (SRTF - Shortest Remaining Time First)** - enak SJF, le da se lahko tekoči proces prekine, če drug pripravljen proces obljublja krajši nemoten tek od preostanka tekočega nemotenega teka
6. **Z zagotovilom** - s pomočjo formule izračuna kvocient k in proces, ki ima najmanjši k dobi CPE. Če vpeljemo še prekinjanje, se k tekom časa povečuje, in ko presega vrednost naslednjega najmanjšega k -ja, se prekine
7. **Loterija** – Razvrščevalnik naključno izbere dovolilnico

29. Na osnovi česa naj vrednotimo in primerjamo algoritme za razvrščanje?

1. Izkoriščenost CPE (CPU utilization)... CPE naj bo čim bolj izkoriščen. V realnosti 40% - 90%
2. Propustnost (throughput)... število končanih procesov v nekem času
3. Čas obdelave (turnaround time)... čas od nastanka procesa do njegovega konca
4. Čakalni čas (waiting time)...
5. Odzivni čas (response time)... čas od nastanka procesa do prve izhodne operacije.
6. Pravičnost (fairness)... vsak proces naj dobi ustrezen čas CPE

30. Kaj pa vpliva na preklopni čas?

1. Hitrost pomnilnika (kjer je shranjen PCB)
2. Število registrov procesorja (ki se morajo shraniti v PCB oz. obnoviti iz njega)
3. (ne)obstoje ukazov za shranitev/obnovo vseh registrov naenkrat
4. Zgradba procesorja
5. Zapletenost komponent OS
6. kakšno je razvrščanje (tj. po katerem algoritmu deluje)
7. ostala morebitna občasna opravila
8. Menjavanje okolja je ozko grlo OS, saj se pri tem lahko izgubi veliko časa.

31. Nit

Je tok izvajanja programa do danega trenutka. Če je sistem enoprocesorski, procesor posveti vsaki niti nekaj časa, tako da vse napredujejo. Če pa je sistem večprocesorski, pa se lahko vsaka nit izvaja na svojem procesorju (to ni paralelno, ampak sočasno). En proces ima lahko več niti. Niti uporabljamo želimo sočasno izvajati več funkcij istega programa ali pa sočasno izvajati več klicev iste funkcije.

32. Podobnosti in razlike med nitmi in procesi

Podobnosti: tudi nit je lahko v stanjih (nova, prekinjena, teče, čaka, končana), tudi nit lahko ustvari nove niti (hčere) ter jih konča in uniči, na enoprocesorskem sistemu tudi niti tekmujejo za procesor. Razlike: komunikacija med istorodnimi nitmi je hitrejša kot med procesi, istorodne niti niso zaščitene ena pred drugo tako kot procesi, ustvarjanje / končanje niti je hitrejša kot pri procesih.

33. Vrste komunikacije

1. Za prenos podatkov skrbi OS (tj. njegova komponenta IPC). Ta način je počasnejši.
2. Za prenos podatkov skrbita proizvajalec in porabnik (preko skupnega pomnilnika)

34. IPC

Je del OS, ki skrbi za komunikacijo in sinhronizacijo med procesi.

Neposredna komunikacija: procesi se pri komuniciranju eksplisitno imenujejo (navedejo).

Simetrična: pošiljatelj in prejemnik drug drugega eksplisitno imenujeta

Asimetrična: samo pošiljatelj eksplisitno navede prejemnika

send(Q,m) ...pošlji procesu Q sporočilo m

receive(x,m) ...prejmi od kogarkoli sporočilo m in potem vrni njegovo ime v spremenljivko x

Posredna komunikacija: procesi imenujejo vmesni medij – nabiralnik. Nabiralnikov je lahko več, vsak ima enolični naslov. Pošiljatelj sporočilo vstavi v nek nabiralnik, od koder ga vzame prejemnik.

Nabiralnik je del procesa P, ki ga je ustvaril. Bere samo P.

Nabiralnik je del OS. Berejo isti katerim P podeli lastništvo. Nevarnost! (tekma za sporočilo, garbage collection)

send(N, m) ...vrzi v nabiralnik N sporočilo m

receive(N,m) ...vzem i iz nabiralnika N sporočilo m

Sinhrono in asinhrono komuniciranje, brez kopičenja, z omejenim kopičenjem, z neomejenim kopičenjem.

Prenašanje kopij sporočila ali reference (pomnilniške naslove).

35. Komunikacija preko skupnega pomnilnika – brez OS

Večja hitrost toda procesa morata sama poskrbeti za komunikacijo.

Proces lahko OS obvesti da drugemu procesu dovoljuje dostop do svojega pomnilnika.

Proces lahko zaprosi za pomnilnik dane velikosti, ki bo skupen. Vsak proces ki ga želi uporabljati si ga mora dodati.

36. Pojasni signale v UNIXu!

Signal je programska prekinitev s strani sistema, drugega programa ali samega sebe. Signal je možno poslati, presteči, prezreti ali obravnavati. Nosijo malo informacije – predvsem za javljanje nenavadnih okoliščin ali napak.

Imenuj razlog, zakaj so signali lahko nezanesljivi.

Signali so lahko nezanesljivi, ker se lahko v izjemnih okoliščinah obnašajo drugače, kot je predvideno.

37. Naštej razlike med vtičnicami in cevmi!

Cevi so enosmerni kanali – za komunikacijo sta potrebni dve, na vtičnice pa se priklopijo dvosmerni kanali. Cevi (razen FIFO, t.j. poimenovane cevi) lahko obstajajo le med dvema procesoma v sorodstvu (otrok-otrok, roditelj-otrok), za vtičnice pa ta relacija ni potrebna.

38. Socketi

Vtičnice so enostaven programski vmesnik za komunikacijo med procesi na istem ali različnih računalnikih. Pred pričetkom komunikacije morata oba procesa P in Q ustvariti vsaj svojo vtičnico. Nato pove vsak svoji vtičnici kakšen naslov bo imela (v mreži je to IP). Sedaj se lahko pogovarjata – strežnik (listen, accept), odjemalec (connect), za oba (send, recv, read, write, close).

39. Razloži kritične odseke / sekcije / dele ! Katere kriterije za splošno rešitev problema

Kritični odsek je del programa oz. procesa, ki dostopa do skupnega sredstva. Bistveni pogoj za rešitev tega problema je da je v kritičnem odseku v vsakem trenutku le en proces.

Kriteriji: medsebojno izključevanje v vsakih okoliščinah, ne sme priti do zastoja, ne sme priti do nepredvideno dolgega odlaganja, izvajanje enega procesa znotraj kritičnega dela ne sme ovirati napredovanja drugih procesov izven kritičnega dela.

- 1) Vzajemno izključevanje procesov: sočasno se lahko izvaja kvečjemu en kritični odsek (tudi če pride do prekinitve)
- 2) Omejeno čakanje na vstop: procesu se ne sme neomejeno dolgo preprečevati vstopa v kr.odsek
- 3) Omejen vpliv na izbor: na to, kdo bo naslednji vstopil v kr. odsek, vplivajo le kandidati za vstop
- 4) Splošnost rešitve: rešitev naj bo neodvisna od števila procesov (n)

40. Razloži semafor!

Semafor je orodje za časovno usklajevanje procesov. Je celoštevilška spremenljivka S, nad katero se izvajajo operacije inicializacija,

wait(S) : while $S \leq 0$ do { no_op; } **S** := S-1; **signal(S)**: **S** := S+1 //atomarno – ne more biti prekinjen

Po navadi sta ti dve operaciji realizirani na OS.

41. Tipi datotek v Linuxu

1. navadna datoteka – /etc/passwd
2. imenik – /bin
3. simbolična povezava – /usr/src/linux
4. cev – mkfifo pipa
5. vtičnica
6. znakovno orientirana naprava – /dev/tty
7. bločno orientirana naprava – /dev/sda

42. Bash ukazi za obdelavo datotek

1. izpis vsebine: cat, less, head, tail
2. urejanje vrstic: sort
3. permutiranje vrstic: shuf
4. odstranjevanje duplikatov: uniq
5. številčenje vrstic: nl
6. obrat vrstic: rev
7. spreminjanje znakov: tr
8. obdelava stolpcev: cut in paste.
9. štetje besed: wc.
10. primerjava vsebine: cmp.

Primer: cat /etc/passwd | cut -d: -f7 | sort | uniq -c | sort -gr