DATOTEČNI SISTEMI

1. Kakšne lastnosti mora imeti sistem za shranjevanje podatkov?  
   2. Predstavi vlogo končnic imen datotek. Kakšne končnice imamo? Katere vrste datotek imajo posamezne končnice?  
   3. Predstavi osnovne organizacijske forme, ki jih tvorijo datoteke. Kako so strukturirani podatki v datotekah?  
   4. Naštej osnovne tipe datotek, ki jih običajno uporablja datotečni sistem.  
   5. Opiši osnovne datoteke in direktorije. Kaj so posebne znakovne datoteke? Kaj so posebne bločne datoteke?  
   6. Predstavi strukturo binarne datoteke. Predstavi strukturo arhivske datoteke.  
   7. Opiši prilastke datotek (file attributes). Predstavi nabolj pogosto uporabljene prilastke datotek.  
   8. Naštej in opiši osnovne operacije nad datotekami.  
   9. Predstavi sekvenčen in direkten dostop do datotek.  
   10. Opiši datoteke preslikane v spomin.  
   11. V katerih primerih je smiselno imeti enonivojski direktorijski sistem?  
   12. Predstavi hierarhični direktorijski sistem.  
   13. Na kakšen način so definirana imena datotek in imena poti na Unix datotečnih sistemih?  
   14. Naštej in opiši operacije na direktorijih.  
   15. Predstavi osnovno arhitekturo datotečnega sistema na disku. Opiši organizacijo particij.  
   16. Predstavi zvezno alokacijo diskovne prostora v datotečnem sistemu.  
   17. Predstavi shranjevanje datotek s povezanim seznamom blokov. Kaj so dobre in kaj slabe lastnosti te predstavitve datotek?  
   18. Opiši alokacijsko tabelo datotek FAT. Na kakšen način je predstavljena datoteka? Kaj so prednosti in slabosti FAT?  
   19. Opiši strukturo i-vozlišča. Kakšne so prednosti i-vozlišča?  
   20. Opiši strukturo zapisov direktorijev. Kako lahko predstavimo direktorije v pomnilniku? Kako so predstavljena imena direktorijev?  
   21. Kako so definirane skupne datoteke datotečnega sistema? Kakšno strukturo tvorijo direktoriji in datoteke v takšnem sistemu?  
   22. Kaj se zgodi, če originalni lastnik zbriše skupno datoteko?  
   23. Obrazloži diagram, ki prikazuje razmerje med hitrostjo prenosa diska in prostorsko učinkovitostjo v odvisnosti od velikosti bloka diska.  
   24. Predstavi načine za beleženje prostih blokov v datotečnem sistemu.  
   ~~25. Recimo da shranjujemo seznam kazalcev na proste bloke v običajnih blokih. Zakaj pride do problemov v primeru, da imamo skoraj poln blok kazalcev in aplikacija konstantno odpira in sprosča začasne datoteke?~~  
   26. Kako datotečni sistem upravlja z diskovnim prostorom uporabnika? Opiši kvote. Katere prilastke ima zapis kvote za omejitev porabe uporabniškega prostora?  
   27. Opiši fizično in logično shranjevanje varnostnih kopij podatkov na disku.  
   ~~28. Predstavi postopek fizičnega shranjevanja varnostnih kopij podatkov diska.~~  
   ~~29. Opiši postopek logičnega shranjevanja varnostne kopije podatkov na disku. Kako lahko uporabimo bitne vektorje za določanje i-vozlišč datotek in direktorijev, ki jih je potrebno shraniti v varnostni kopiji?~~  
   30. Kako lahko preverimo konsistentnost datotečnega sistema?  
   31. Opiši postopek preverjanja konsistentnosti datotečnega sistema z uporabo bitnih vektorjev blokov. V kakšnih stanjih je lahko datotečni sistem? Opiši posamezno stanje.  
   32. Opiši strukturo vmesnega pomnilnika blokov diska. Kakšne vrste strani lahko identificiramo? Je LRU strategija primerna za vse vrste strani?  
   33. Kako lahko organiziramo shranjevanje i-vozlišč fizično na disku? Komentiraj različne arhitekture shranjevanje i-vozlišč: na začetku diska, porazdeljeno po skupinah cilindrov.  
   34. Opiši dnevniško strukturiran datotečni sistem.

34. Dnevniški datotečni sistem je datotečni sistem, ki spremlja spremembe, ki še niso bile izvedene v glavnem delu datotečnega sistema, tako da beleži cilj takih sprememb v podatkovni strukturi, znani kot "dnevnik", ki je običajno krožni dnevnik.

21. S podporo za skupni datotečni sistem so vsi datotečni sistemi, ki jih priklopi sistem, ki sodeluje v skupnem datotečnem sistemu, na voljo vsem sodelujočim sistemom. Z drugimi besedami, ko sodelujoči sistem priklopi datotečni sistem, je ta datotečni sistem dostopen kateremu koli drugemu sodelujočemu sistemu.

22. Izbrišite datoteke in mape v skupni rabi, ko ste lastnik: Če ste lastnik datoteke v skupni rabi in izbrišete isto datoteko v skupni rabi, ki je v vaši lasti, bo datoteka vseh sodelavcev v skupni rabi popolnoma izbrisana iz pogona. In ne bodo mogli več dostopati do dokumenta.

26. Datoteke so običajno shranjene na disku, zato je upravljanje prostora na disku velika skrb oblikovalcev datotečnega sistema. Za shranjevanje n-bajtne datoteke sta možni dve splošni strategiji: dodeli se n zaporednih bajtov prostora na disku ali pa se datoteka razdeli na več (ne nujno) sosednjih blokov.

20. Struktura imenika je organizacija datotek v hierarhijo map. Imeniki so organizirani kot obrnjena drevesna struktura. Samo en imenik, na vrhu drevesa, ni vsebovan v nobenem drugem imeniku. To se imenuje korenski imenik, njegovo ime pa je predstavljeno s poševnico (/).

27. Fizično shranjevanje diska – Ukaz dump: shranjevanje sektorjev 0..N • Lahko zelo hitra operacija – Dostop do prostih/zasedenih blokov pohitri shranjevanje • Logično shranjevanje varnostnih kopij – Od korenskega direktorija rekurzivno po vseh datotekah – Shranimo samo spremenjene datoteke? • Inkrementalno kopiranje ne omogoča restavracijo izbrisanih datotek

5.

Drug pomemben koncept v UNIX-u je posebna datoteka. Na voljo so posebne datoteke, da so V/I naprave videti kot datoteke. Tako jih je mogoče brati in pisati z istimi sistemskimi klici, kot se uporabljajo za branje in pisanje datotek. Obstajata dve vrsti posebnih datotek: blokovne posebne datoteke in posebne znakovne datoteke. Posebne datoteke blokov se uporabljajo za modeliranje naprav, ki so sestavljene iz zbirke naključno naslovljivih blokov, kot so diski. Datoteke s posebnimi znaki so povezane z vhodom/izhodom in se uporabljajo za modeliranje serijskih V/I naprav, kot so terminali, tiskalniki in omrežja. Za modeliranje diskov se uporabljajo posebne blokovne datoteke. V tem poglavju nas bodo zanimale predvsem običajne datoteke.

19.

Glede na i-vozlišče je nato mogoče najti vse bloke datoteke. Velika prednost te sheme pred povezanimi datotekami, ki uporabljajo tabelo v pomnilniku, je, da mora biti i-vozlišče v pomnilniku le, ko je ustrezna datoteka odprto. Če vsak inode zaseda n bajtov in je lahko naenkrat odprtih največ k datotek, je skupni pomnilnik, ki ga zaseda matrika, ki vsebuje i-vozlišča za odprte datoteke, samo kn bajtov. Le toliko prostora je treba rezervirati vnaprej. Ta niz je običajno veliko manjši od prostora, ki ga zaseda tabela datotek, opisana v prejšnjem razdelku. Razlog je preprost. Tabela za shranjevanje povezanega seznama vseh diskovnih blokov je sorazmerna s samim diskom. Če ima disk n blokov, tabela potrebuje n vnosov. Ko diski rastejo, se ta tabela linearno povečuje z njimi. Nasprotno pa shema i-node zahteva polje v pomnilniku, katerega velikost je sorazmerna z največjim številom datotek, ki so lahko odprte hkrati. Ni pomembno, ali je disk velik 100 GB, 1000 GB ali 10.000 GB.

6. Struktura. Binarne datoteke se običajno obravnavajo kot zaporedje bajtov, kar pomeni, da so binarne števke (biti) združene v osmice. Binarne datoteke običajno vsebujejo bajte, ki naj bi jih interpretirali kot nekaj drugega kot besedilne znake. V računalništvu je arhivska datoteka računalniška datoteka, ki je sestavljena iz ene ali več datotek skupaj z metapodatki. Arhivske datoteke se uporabljajo za zbiranje več podatkovnih datotek skupaj v eno datoteko za lažjo prenosljivost in shranjevanje ali preprosto za stiskanje datotek, da porabijo manj prostora za shranjevanje.

SPOMIN

1. Predstavi shemo spomina v mono-programskem sistemu.

2. Predstavi shemo spomina v multi-programskem sistemu, ki uporablja

particije.

3. Kaj je to stopnja multiprogramiranja?

~~4. Nariši graf, ki opiše odvisnost izrabe procesorskega časa~~

~~od stopnje multiprogramiranja~~.

5. Zakaj je potrebna relokacija in zaščita pomnilniškega prostora,

ki ga uporabljajo programi?

6. Kaj je to izmenjevanje spomina? Zakaj ga uporabljamo?

7. Predstavi podatkovne strukture za predstavitev prostega

pomnilnika v multi-programskem sistemu.

8. Naštej funkcije MMU.

9. Definiraj virtualni pomnilnik realiziran z ostranjevanjem.

10. Nariši shemo virtualnega pomnilnika in RAM pri uporabi

ostranjevanja.

~~11. Predstavi tabelo strani, ki jo uporabljamo pri ostranjevanju.~~

~~Kaj če je prevelika za RAM? Opiši zapis tabele strani.~~

12. Opiši TBL - izravnalnik za prevajanje virtualnih naslovov

v realne naslove. Kako je realiziran?

13. Zakaj uporabljamo invertirane tabele strani? Predstavi

podatkovno strukturo invertirane tabele strani. Predlagaj

načine implementacije.

14. Naštej algoritme za izmenjavo strani virtualnega pomnilnika.

15. Napiši pseudo kodo za algoritem "druga možnost". Kako algoritem

16. Predstavi implementacije algoritma Najzadnje uporabljen.

17. Opiši algoritem Delovna množica. Predlagaj načine implementacije

algoritma.

18. Napiši pseudo kodo za algoritem staranja.

19. Kaj je to Belady-jeva anomalija?

20. Opiši skladovne algoritme za izmenjavo strani.

21. Kako bi izbrali velikost strani virtualnega pomnilnika?

22. Predlagaj postopek za ciklično izvajanje čiščenja pomnilnika.

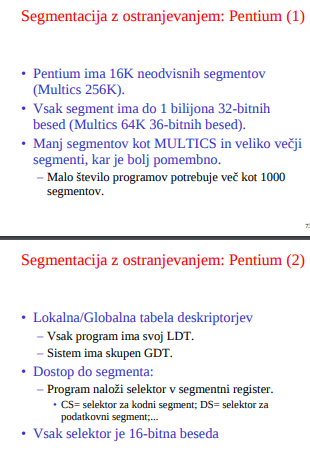
23. Opiši razloge za uporabo segmentacije.

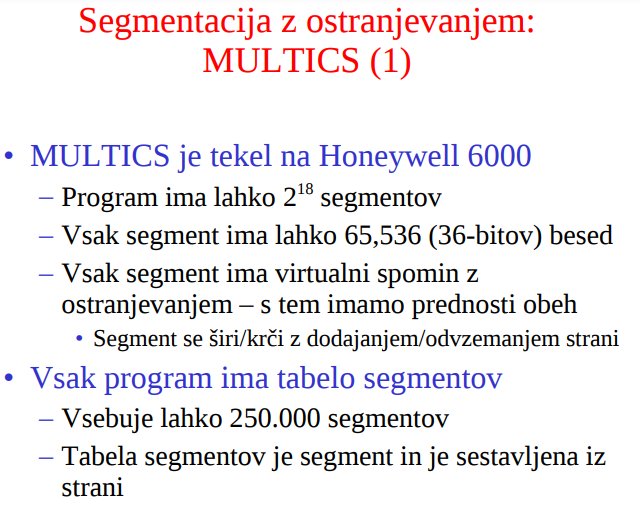
24. Predstavi načine za realizacijo segmentacije.

25. Opiši implementacijo segmentacije z ostranjevanjem na MULTICS.

26. Predstavi segmentacijo z ostranjevanjem na Pentium.

26.



25. 

24. • Implementacija čiste segmentacije: – Segmenti so variabilne velikosti - strani niso – Spomin se fragmentira, ko se segmenti zamenjujejo – Segmente lahko stisnemo skupaj periodično

23. Segmentacija pomaga tržnikom, da so bolj učinkoviti v smislu časa, denarja in drugih virov. Segmentacija trga omogoča podjetjem, da spoznajo svoje stranke. Pridobijo boljše razumevanje strankinih potreb in želja ter tako lahko prilagodijo oglaševalske akcije segmentom kupcev, ki bodo najverjetneje kupili izdelke.

Prednosti segmentacije: – Variranje velikosti segmentov – Poenostavljeno povezovanje in vlaganje programov (n,0) – Procedure se izvajajo na naslovu 0 – Skupni segmenti (knjižnice, grafika, ...) – Zaščita (R,X,RX,...) segmenta – pomaga pri iskanju napak – Programer ima iluzijo, da so vsi segmenti vedno v pomnilniku

22. Potreba po procesih v ozadju, deamon za ostranjevanje – Periodično pregleda stanje spomina • Ko je premalo okvirjev prostih – Izbere strani za odstranitev z algoritmom za zamenjavo strani • Lahko uporabimo krožni seznam (ura) – Kot običajen algoritem za zamenjavo strani

19. V računalniškem pomnilniku je Béládyjeva anomalija pojav, pri katerem povečanje števila okvirjev strani povzroči povečanje števila napak strani za določene vzorce dostopa do pomnilnika. Ta pojav se običajno pojavi pri uporabi algoritma za zamenjavo strani FIFO (first-in-first-out).

NITI IN PROCESI

1. Opiši procesni model za implementacijo programskih sistemov.  
   Kaj je proces? Kaj je multiprogramiranje?  
2. Naštej dogodke, kiso vzrok za glavne spremembe stanja procesa.  
   Kdaj se proces kreira? Kdaj se konča?  
3. Kakšno strukturo lahko tvori množica procesov? Opiši relacije  
   med procesi.  
4. V katerih stanjih je lahko proces? Nariši diagram prehajanja stanj  
   procesa.  
5. Naštej podatke, ki jih hrani zapis procesne tabele.  
6. Opiši kaj se zgodi ob prekinitvi procesa. Katere akcije se  
   zgodijo ob prekinitvi procesa v izvajanju in zagonu novega procesa?  
7. Opiši model niti. Kakšna je relacija med nitmi in procesi?  
8. Predstavi podatke, ki jih hrani nit. Kateri podatki so predstavljeni  
   v okviru procesa in kateri v okviru niti?  
9. Predstavi nekaj primerov programskih sistemov, ki uporabljajo niti.  
   Opiši v grobem arhitekturo primerov programskih sistemov.  
10. Opiši implementacijo niti v uporabniškem prostoru. Kako je  
    implementirana kontrola prenosa izvajanja med nitmi enega procesa?  
11. Opiši posledice implementacije niti v jedru.  
12. Kdaj se lahko pojavi stanje tekmovanja pri komunikaciji med  
    dvemi ali večimi procesi?  
13. Opiši pojma kritično področje in vzajemno izključevanje pri  
    izvajanju medprocesne komunikacije. Naštej štiri pogoje za  
    omogočanje vzajemnega izključevanja.   
14. Opiši vzajemno izključevanje s čakanjem v zanki. Predstavi programske  
    rešitve, ki  jih lahko uporabljamo za kontrolo dostopa do kritične  
    sekcije.  
15. Napiši pseudokodo za vsako od predstavljenih programskih rešitev  
    za dostop do kritične sekcije: prepoved prekinitev, ključavnice,  
    striktno izmenjevanje, Dekkerjev algoritem, Petersonov algoritem in  
    TSL (test­and­set­lock).  
15. Opiši vzajemno izključevanje s spanjem in zbujanjem.  
16. Zakaj uporabljamo semaforje? Napiši pseudokodo za implementacijo  
    števnega semaforja.  
17. Kaj je to mutex? Kakšna je razlika med semaforjem in mutexom?  
18. Opiši delovanje monitorja. Kako so monitorji implementirani v Javi?  
19. Definiraj problem proizvajalca in porabnika. Napiši pseudokodo  
    za reševanje problema proizvajalec-uporabnik z uporabo:  
    a) semaforjev, b) mutex in c) monitorja.  
20. Opiši medprocesno komunikacijo z uporabo sporočil.  
21. Kako je definiran gradnik za sinhronizacijo procesov z imenom  
    meja. Podaj nekaj primerov uporabe meje v praksi.  
22. Definiraj problem obedujočih filozofov. Zapiši eno izmed možnih  
    rešitev problema, za katero se lahko zgodi smrtni objem. Kako se zgodi  
    smrtni objem? Predstavi možne rešitve za izogibanje smrtnem objemu        
    pri problemu filozofov.  
23. Predstavi problem bralcev in pisalcev: več različnih bralcev lahko  
    hkrati bere kritične podatke, med pisanjem pa mora pisalec dobiti  
    ekskluziven dostop do kritične sekcije.  
~~24. Definiraj problem spečega brivca z definicijo podatkovnih struktur  
    in postopkov, ki modelirajo dogajanje v brivnici. Preveri ali je možno,  
    da pride do smrtnega objema.~~  
25. Kaj so osnovni kriteriji za razporejanje procesov po tipih sistemov:  
    za vse sisteme, sveženjske in interaktivne sisteme?  
26. Predstavi sveženjsko razporejanje. Opiši naslednje metode za  
    sveženjsko razporejanje: prvi pride prvi melje, najprej krajši posel,  
    in najkrajši preostanek časa najprej.  
~~27. Opiši sveženjski sistem, ki ima tri razporejevalnike: razporejanje  
    vstopa v izvajanje na sistemu, razporejanje CPU po procesih  
    in razporejanje procesov z začasnim umikanjem procesov na disk.~~28. Predstavi razporejanje v interaktivnih sistemih. Kaj so pomembni  
    kriteriji? Kako se kriteriji v sveženjskih sistemih razlikujejo od  
    kriterijev v interaktivnih sistemih.  
29. Opiši algoritem za razporejanje v interaktivnih sistemih Round-Robin.  
    Komentiraj implementacijo algoritma. Katere podatkovne strukture  
    lahko uporabimo?    
30. Predstavi prioritetno razporejevanje v interaktivnih sistemih.  
31. Opiši model razporejanja v sistemih v realnem času.  
32. Opiši razliko med politiko in mehanizmom razporejanja procesov.

SMRTNI OBJEMI

1. Predstavi računalniške vire iz stališča procesov in  
   sinhronizacije procesov.

2. Podaj definicijo smrtnega objema.

3. Naštej pogoje, ki so potrebni, da lahko pride do  
   smrtnega objema.  
4. Definiraj model za predstavitev procesov, virov  
   in smrtnih objemov. Kako ugotovimo, da je prišlo do  
   smrtnega objema?  
5. Opiši štiri strategije za delo s smrtnim objemom,  
   ki so bile predstavljene na predavnjih.  
6. Opiši nojev algoritem za reševanje iz smrtnega objema.  
   Kdaj je nojev algoritem smiseln?  
7. Predstavi model in algoritem za odkrivanje smrtnih  
   objemov, če imamo en sam vir določenega tipa.  
8. Predstavi model in algoritem za odkrivanje smrtnih objemov,  
   če imamo več virov danega tipa.  
9. Predstavi metode za reševanje iz smrtnega objema in  
   komentiraj posledice izbire metode. V kakšnih pogojih  
   je dana metoda sprejemljiva?  
10. Kako si lahko predstavimo izogibanje smrtnem objemu  
    s trajektoriji virov?  
11. Opiši model za predstavitev stanja sistema, ki ima  
    K primerkov vira enega tipa in N procesov, ki dostopajo  
    do teh K-tih virov. Poznamo obstoječe stanje sistema  
    (kateri procesi zasedajo koliko virov) in potrebe procesov  
    po virih.  
12. Predstavi bankirjev algoritem za en vir.  
13. Predstavi bankirjev algoritem za več virov.  
14. Opiši vsako posamezno metodo za onemogočanje smrtnega objema,  
    je bila predstavljena za posamezen napad na pogoje, ki so  
    potrebni za pojavitev smrtnega objema.  
15. Predstavi postopek dvo-faznega zaklepanja.

8. PREZENTACIJA srti objemi slajd 13

12 i 13. prezentacija 21 alajd

UVOD

1. Kaj je operacijski sistem?  
2. Opiši na kratko zgodovino operacijskih sistemov.  
3. Predstavi osnovne komponente računalniškega sistema,  
   ki ga nadzira operacijski sistem.  
4. Kakšne vrste operacijskih sistemov imamo?  
5. Povzami osnovne značilnosti modernih mikro-procesorjev.  
6. Na kratko predstavi delovanje dinamičnega pomnilnika RAM.  
7. Nariši shemo osnovnih komponent računalniškega sistema in  
   vodil, ki jih povezujejo.  
8. Predstavi datotečni sistem. Kaj ga sestavlja? Kakšne strukture  
   tvorijo komponente datotečnega sistema?  
9. Predstavi osnovne koncepte izvajanja programov na operacijskem sistemu.  
   Kaj je to proces, cevovod, medprocesna komunikacija, nit, ... ?  
10. Opiši klic sistemske rutine. Kaj se zgodi v operacijskem sistemu  
    ob sistemskem klicu?  
11. Naštej nekaj klicev za delo z datotekami, za delo s spominom,  
    za delo s procesi in za delo z vhodno-izhodnimi napravami.  
12. Napiši pseudokodo za enostavno lupino operacijskega sistema.  
13. Opiši model Odjemalec-Strežnik. Kako porazdeljeni sistemi  
    uporabljajo predstavljeni model?    
14. Predstavi sturkturo operacijskega sistema. Kaj so najbolj pomembne  
    komponente OS? Kako so med sabo povezane?

VHODNO/IZHODNE NAPRAVE

1. Opiši krmilnik naprave. Kakšne funkcije ima?  
2. Predstavi programiranje V/I naprav s pomočjo preslikave  
   spomina V/I naprav.  
3. Predstavi programiranje V/I naprav z DMA krmilnikom.  
4. Opiši vlogo prekinitev pri delu z V/I napravami.  
5. Naštej principe in cilje moderne V/I programske opreme.  
6. Predstavi programske tehnike za implementacijo prenosa  
   podatkov med V/I napravo in pomnilnikom.  
7. Napiši skico implementacije prenosa podatkov med V/I  
   napravo in pomnilnikom z uporabo DMA krmilnika.  
8. Predstavi programske nivoje V/I programske opreme.  
9. Opiši prekinitveni program, ki dela z V/I napravo.  
   Kako poteka prenos podatkov?  
10. Opiši gonilnik V/I naprave. Kakšno pozicijo ima  
    v V/I programski opremi?  
11. Kako naredimo V/I programsko opremo neodvisno od naprav?  
12. Predstavi vlogo izravnalnika v V/I programski opremi.  
13. Opiši nivoje V/I programske opreme in funkcije  
    posameznega nivoja.  
14. Predstavi arhitekturo modernega diska.  
15. Opiši razliko med fizično in navidezno geometrijo diska.  
16. Kaj je to RAID? Kakšne nivoje ima? Predstavi način zapisa  
    podatkov za vsak posamezen RAID nivo.  
17. Opiši arhitekturo CD-ROM diska. Opiši zapis podatkov in  
    logični načrt podatkov na CD-ROM.  
18. Predstavi CD-R, CD/RW in DVD. Opiši izboljšave glede na  
    CD-ROM.  
19. Opiši postopek formatiranja diska. Kakšne korake vsebuje      
    formatiranje? Predstavi fizično in logično formatiranje.  
20. Definiraj problem razporejanja opravil za diskovno roko.  
21. Zapiši pseudokodo algoritmov za razporejanje opravil diskovne  
    roke: a) najprej najkrajša pot in b) dvigalo.

19. Formatiranje diska je postopek konfiguriranja medija za shranjevanje podatkov, kot je trdi disk, disketa ali bliskovni pogon, za prvo uporabo. Vse obstoječe datoteke na pogonu bi bile izbrisane s formatiranjem diska. Formatiranje diska se običajno izvede pred prvo namestitvijo ali pred namestitvijo novega operacijskega sistema.

17. prez 63

12. Izenačevalnik (imenovan tudi »EQ«) je zvočni filter, ki izolira določene frekvence in jih poveča, zniža ali pusti nespremenjene. Izenačevalnike najdemo na številnih elektronskih napravah.