Operatsioonisüsteemi struktuur

Ülevaade

- Süsteemi komponendid
- Operatsioonisüsteemi teenused
- Süsteemifunktsioonid
- Süsteemsed programmid
- Süsteemsed struktuurid
- Virtuaalmasinad
- Süsteemi disain ja realiseerimine
- Süsteemi genereerimine

Levinumad süsteemi komponendid

- Protsessihaldus
- Põhimälu haldus
- Failide haldus
- I/O süsteemi haldus
- Sekunaadrse salvestusruumi haldus
- Võrgu tugi
- Kaitsesüsteem
- Käsuinterpretaator

Protsessihaldus

- Protsess on täitmisel olev programm
- Protsess vajab oma tööks ressursse (protsessoriaeg, mälu, failid, I/O seadmed)
- Operatsioonisüsteem vastutab protsessihalduses järgmiste tegevuste eest:
  - Protsesside loomine ja hävitamine
  - Protsesside peatamine ja jätkamine
  - Protsesside sünkroniseerimise mehhanismid
  - Protsessidevahelise side mehhanismid
  - Tupikute haldamise mehhanismid

Põhimälu haldus

- Mälu on suur massiiv sõnadest või baitidest, igal oma aadress. Kiiresti kättesaadav protsessori ja I/O seadmete poolt.
- Põhimälu läheb kaotsi masina rikke või välja lülitamise korral
- Operatsioonisüsteem vastutab mäluhalduses järgmiste tegevuste eest:
  - Arve pidamine selle üle, missugused mälualad on hetkel kasutuses ja kelle poolt
  - Otsustamine, missuguseid protsesse mälu vabanedes mällu laadida
  - Mälu hõivamine ja vabastamine, kui protsessid seda küsivad

Failihaldus

- Fail on kogumik infot omavahel seotud asjadest
- Konkreetse faili struktuuri defineerib faili looja
- Enamasti hoitakse failides programme (nii lähtetekstina kui binaarkujul) ja andmeid
- Operatsioonisüsteem vastutab failihalduses järgmiste tegevuste eest:
  - Failide loomine ja kustutamine
  - Kataloogide loomine ja kustutamine
  - Tugi failide ja kataloogidega manipuleerimise primitiividele
  - Failide sekundaarsetele salvestusseadmetele jagamine
  - Failide varukoopia tegemise organiseermine

I/O süsteem

- I/O süsteem koosneb:
  - I/O puhvrite süsteemist
  - Vahemälu (*cache*) haldamise süsteemist
  - Üldisest seadmedraiverite liidesest
  - Konkreetsete seadmete draiveritest

Sekunaadrse salvestusruumi haldus

- Põhimälu ei säili pärast voolukatkestust ning ei mahuta kõiki vajaminevaid programme ja andmeid
- Seetõttu tuleb kasutada sekundaarset salvestusruumi, et vajalik info säiliks
- Kettad jne
- Operatsioonisüsteem vastutab failihalduses järgmiste tegevuste eest:
  - Vaba ruumi üle arve pidamine
  - Ruumi jagamine
  - Kettapöörduste planeerimine

Kaitsemehhanismid

- Kaitsemehhanismid kontrollivad juurdepääsu süsteemi ja kasutaja ressurssidele programmide, protsesside või kasutajate poolt
- Kaitsemehhanism peab:
  - Eristama autoriseeritud ja autoriseerimata kasutajaid
  - Määrama juurdepääsureeglid
  - Need reeglid kehtestama

Käsuinterpretaator

- Palju käske, mis operatsioonisüsteemile antakse, on operatsioonisüsteemi endaga seotud juhtimiskäsud:
  - Protsesside loomine
  - I/O operatsioonid
  - Põhimälu haldamine
  - Failisüsteemi manipuleerimine
  - Kaitsemehhanismid
  - Võrguoperatsioonid
- Käsuinterpretaator – programm süsteemi juhtimiseks, mis suhtleb kasutajaga
- Näide: UNIXi shell, MS-DOS'i COMMAND.COM, MacOSi GUI+Finder, Windowsi GUI+Explorer

Operatsioonisüsteemi teenused

- Teenused, mida operatsioonisüsteem pakub programmidele
- Programmide täitmine (programmi mällu laadimine ja käivitamine)
- I/O operatsioonid – kuna programmid ei saa enamasti otse välisseadmete poole pöörduda, peab OS selleks teenuseid pakkuma
- Failisüsteemi manipuleerimine – failide loomine, kustutamine, lugemine, kirjutamine, kataloogioperatsioonid
- Side – andmete vahetamine erinevate protsesside vahel (samas arvutis või erinevates arvutites), realiseeritakse teadete saatmisega või jagatud mäluga
- Vigade avastamine – protsessori, mälu, I/O seadmete ja kasutajaprogrammide vigade avastamine ja neile adekvaatselt reageerimine

Operatsioonisüsteemi lisafunktsioonid

- Mitte kasutajale orienteeritud, vaid süsteemi enda paremaks funktsioneermiseks
- Ressursside hõivamine ja vabastamine
- Arvepidamine – kes kuipalju ja missuguseid ressursse on kasutanud (arvete esitamiseks, statistika jaoks)
- Kaitse ja turvalisus

Süsteemifunktsioonid

- Süsteemifunktsioonid on liideseks töötava programmi ja operatsioonisüsteemi tuuma vahel
- Enamasti eksisteerivad assemblerkeele tasemel
- Tihti otse saadaval ka süsteemprogrammeerimise keeltes (C, C++)
- Kolm parameetrite edastamise viisi:
  - Parameetrid registrites
  - Parameetrid mälutabelis, fikseeritud registris tabeli aadress
  - Parameetrid magasinis

Süsteemifunktsioonide levinumad tüübid

- Protsessihaldus
- Failihaldus
- Seadmete haldus
- Üldinfo
- Side

Süsteemsed programmid

- Aitavad luua mugavat keskkonda rakendusprogrammide jooksutamiseks ja arenduseks
- Annavad juurdepääsu süsteemifunktsioonidele ja teenustele kasutajale kättesaadaval kujul
- Kasutaja vaade süsteemile on enamasti defineeritud just süsteemsete programmide kaudu
- Tuuma vaatepunktist on tegu harilike rakendusprogrammidega
  - Failide manipuleerimiseks ja modifitseerimiseks
  - Staatusinfo jagamiseks
  - ...
- Osa süsteemseid programme on vahel realiseeritud käsuinterpretaatori osana

Struktuur: MS-DOS

- Kirjutatud nii, et saada maksimaalset funktsionaalsust vähese mälukasutusega
- Pole mooduliteks jagatud
- Mõningane sisemine struktuur olemas, kuid liidesed ja funktsionaalsus pole eraldatud
- Rakendused laiendavad opsüsteemi nagu heaks arvavad

Struktuur: originaalne UNIX

- Originaalne UNIX oli samuti piiratud riistvara funktsionaalsusega
- Natuke struktuurne
- Kaks eraldiseisvat osa: süsteemprogrammid ja tuum
- Tuum on kõik, mis jääb allapoole süsteemifunktsioonide liidest
- Tuum tegeleb mäluhalduse, protsessoriaja jagamise, failisüsteemide halduse, seadmedraiverite ja muude opsüsteemi funktsioonidega — kõik ühel tasandil
- Endiselt monoliitne süsteem

Kihiline lähenemine

- Opsüsteem on jagatud mingiks arvuks kihtideks (tasemeteks)
- Iga kiht kasutab eelmiste kihtide teenuseid
- Alumine kiht on riistvara
- Ülemine kiht on kasutajaliides
- Modulaarsus: iga kiht kasutab ainult alumiste kihtide operatsioone ja teenuseid
- Näide: OS/2

Mikrokernel

- Võimalikult suur osa tuumast tõstetakse moodulitena kasutaja tasemele
- Moodulite vahel käib side teadete edastamise abil
- Plussid:
  - Lihtsam laiendada
  - Lihtsam uuele arhitektuurile portida
  - Töökindlam (vähem koodi töötab kerneli moodis)
  - Turvalisem
- Näited: Tru64 UNIX, MacOS X, QNX, natuke ka Windows NT

Virtuaalmasinad

- Kihilise süsteemi loogiline jätk
- Kõiki allapoole jäävaid kihte vaadeldakse lihtsalt kui riistvara, mis antud kihil kasutada on
- Virtuaalmasin annab klientmasinatele tegeliku riistvaraga identse liidese
- Tekitab illusiooni paljudest protsessidest, millest igaüks jookseb oma protsessoril, jooksutab seal oma operatsioonisüsteemi oma (virtuaal-) mäluga
- Selline klient-opsüsteem võib olla lihtne ühekasutajaoperatsioonisüsteem
- IBM üks põhitegijaid

Virtuaalmasinad

- Füüsilise masina ressursse jagatakse virtuaalmasinate loomiseks
- Protsessori(te) jagamine jätab mulje, et kasutajatel on oma protsessorid
- Failisüsteem ja spuulimine annavad virtuaalsed kaardilugejad, kettad ja printerid
- Harilik ajajaotussüsteemi kasutav terminal on virtuaalmasina konsooliks
- Jagatud kettad (*minidisk*)
- Virtuaalsed võrguliidesed virtuaalmasinate vahel suhtlemiseks

Virtuaalmasinate head ja vead

- Täielik kaitse süsteemi ressurssidele virtuaalmasinate isoleerituse tõttu
- Ressursse seetõttu otse jagada ei saa
- Teoreetiliselt võimas vahend opsüsteemide uurimiseks ja arendamiseks
- Keeruline realiseerida, kuna riistvara täpse koopia emuleerimine on raske ja vajab efektiivseks tööks spetsiaalset riistvara (täielikult virtualiseeritav protsessor)

Java virtuaalmasinad

- Java programmid kompileeritakse platvormist sõltumatusse baitkoodi
- Virtuaalmasin (JVM) täidab baitkoodi
- JVM koosneb
  - Klassi laadurist
  - Klassi verifitseerijast
  - Interpretaatorist (harilikud interpretaatorid ja JIT-kompilaatorid)

Eksokernelid

- Virtuaalmasinate edasiarendus
- Seadmete pöördusi ei tõlgita (enam pole igal klientsüsteemil oma 0-aadressi)
- Erinevatele protsessidele antakse erinevad aadressivahemikud
- Nendest aadressivahemikest kinni pidamist kontrollitakse tuuma tasemel
- Vähem lisavõhma tõlkimise peale

Süsteemi disain

- Disaini eesmärgid:
  - Kasutajale suunatud eesmärgid – opsüsteem peab olema mugav kasutada, lihtne õppida, töökindel, turvaline ja kiire
  - Süsteemi enda jaoks vajalikud eesmärgid – opsüsteemi peab olema lihtne disainida, realiseerida, hooldada, ta peab olema paindlik, töökindel, veavaba ja efektiivne
- Mehhanismid – kuidas midagi teha
- Poliitika – mida teha
- Mehhanismid ja poliitika on soovitav lahus hoida
- Assembler vs kõrgkeel, teisaldatavus

Süsteemi genereerimine

- Igas konkreetses masinas on oma riistvarakonfiguratsioon ja kasutamiseesmärk
- Kuidas genereerida levitatavast opsüsteemist antud masinale vastav konfiguratsioon?
- Mis protsessori(id) ja käsustik?
- Palju mälu?
- Mis välisseadmed?
- Missugused operatsioonisüsteemi parameetrid?
- Lahendusi: lähtekoodi muutus, ümberlinkimine, käigu pealt tabelite järgi ringi konfigureerimine
- Genereeritud süsteem tuleb lõpuks ka alglaaditavaks teha