

Operatsioonisüsteemid

LTAT.06.001

Meelis Roos

mroos@ut.ee

Arvutiteaduse Instituut

Tartu Ülikool

sügis 2020

Kirjandus

- A. S. Tanenbaum, H. Bos. Modern Operating Systems. 4th ed. Prentice Hall. 2015.
 - sobib ka 3rd edition
- A. Silberschatz, P. B. Galvin, G. Gagne. Operating System Concepts. 6th ed. (või uuem) John Wiley & Sons. 2002.
- G. Nutt. Operating Systems: A Modern Perspective. 3rd ed. Addison-Wesley. 2003.
- J. Vendelin, Operatsioonisüsteemid, 2003
- K. Taurus, Operatsioonisüsteemid. Praktikumijuhend, 2007

Veeb

- Kursuse veebileht:
<https://courses.cs.ut.ee/2020/os/>
- Loengute slaidid ja videosalvestused
- Praktikumijuhendid
- Kodused ülesanded ja osalejate lahendused esitatakse üldjuhul ülaltoodud veebilehel.

Eesmärgid

- Kursuse läbinu mõistab, mis ja milleks on operatsioonisüsteem ning kuidas see töötab
- Kursuse läbinu on paigaldanud ja skriptinud vähemasti Windows- ja Unix-tüüpi opsüsteeme

Kursuse läbinu:

- Saab aru, missugustest kihtidest ja komponentidest operatsioonisüsteem koosneb ning kuidas toimub suhtlus erinevate osade vahel ning füüsilise ja virtualiseeritud riistvaraga
- Mõistab paralleeltöötlust
- Saab aru mäluhaldusest
- Saab aru salvestussüsteemidest ja failisüsteemidest
- Saab aru arvutivõrkude kihtidest ja TCP/IP protokollipere kihtidest
- On paigaldanud ja hallanud Windows- ja Unix tüüpi operatsioonisüsteeme
- Oskab kasutada skriptimist operatsioonisüsteemi tasemel objektide juhtimise automatiseerimiseks

Kursuse läbimise eeldused

- Pühendumine ainele iga nädal
- Oskus teha iseseisvat tööd
- Oskus esitada oma teadmisi arusaadavalt
- Oskus vormistada oma töö UT tudengile kohaselt

Kursuse läbimise tingimused

- 30% punktidest tuleb kirjalikust eksamist
- 45% punktidest tuleb praktikumitööde õigeaegsest esitamisest
- 25% punktidest tuleb kodustest töödest
(programmeerimisülesanded OS simuleerimisest)
- Eksamile pääsemiseks on vaja saada vähemalt pooled punktid kodutöödest ja praktikumidest kokku
- Eksamitööst on vaja saada vähemalt 15 punkti 30-st

Kursuse kava

- Arvutisüsteemi ja operatsioonisüsteemi struktuur
- Protsesside haldus
 - Protsessid ja lõimed, protsessoriaja jagamine
- Mäluhaldus
 - Põhimäluhaldus, virtuaalmälu
- Salvestussüsteemid
- Failisüsteem
 - Failisüsteemi liides ja realisatsioon
- Protsessidevaheline suhtlus
 - Sünkroniseerimine, tupikud
- Arvutivõrgu kihiline mudel
 - TCP/IP

Operatsioonisüsteemi komponendid

- Protsessihaldus
- Põhimäluhaldus
- I/O süsteemi haldus
- Sekundaarse salvestusruumi haldus
- Failide haldus
- Võrgu tugi
- Kaitstesüsteem
- Kasutajaliides

Protsessihaldus

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

Mäluhaldus – mis on mälu?

- **Siin tegeleme põhimäluga (RAM – *Random Access Memory*)!**
- Mälu on suur massiiv sõnadest või baitidest, igal oma aadress; kiiresti kättesaadav protsessori ja I/O seadmete poolt
- Põhimälu sisu läheb kaotsi masina rikke või väljalülitamise korral
- Mitte segamini ajada sekundaarse salvestusruumiga (*secondary storage*)
- Vett segav erand: säiliv põhimälu (*non-volatile RAM, NVDIMM*)
 - Vajab operatsioonisüsteemi erituge

Mäluhaldus

- Operatsioonisüsteem vastutab mäluhalduses järgmiste tegevuste eest:
 - Protsessidele oma aadressiruumide andmine
 - Mälu hõivamine ja vabastamine, kui protsessid seda küsivad
 - Arve pidamine selle üle, millised mälu alad on hetkel kasutuses ja kelle poolt
 - Otsustamine, missuguseid protsesse mälu vabanedes mällu laadida
- Oma aadressiruumi piires mälupaigutuse korraldamine on protsessi enda ülesanne
- Protsess kasutab selleks tüüpiliselt opsüsteeme poolt pakutavaid teeke

Failihaldus

- Fail on kogumik infot omavahel seotud asjadest
- Failid on harilikult salvestatud sekundaarsesse salvestusruumi
- 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 organiseerimine

I/O süsteem

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

Sekundaarse salvestusruumi haldus

- Põhimälu ei säili pärast volukatkestust ning ei mahuta kõiki vajaminevaid programme ja andmeid
- Seetõttu tuleb vajaliku info säilimiseks kasutada sekundaarset salvestusruumi
- Kettad, SSD, magnetlindid, NVRAM jne
- Operatsioonisüsteem vastutab sekundaarse salvestusruumi halduses järgmiste tegevuste eest:
 - Vaba ruumi üle arvepidamine
 - Ruumi jagamine
 - Kettapöörduste planeerimine

Kaitsemehhanismid

- Kaitsemehhanismid kontrollivad juurdepääsu süsteemi ja kasutaja ressursidele programmide, protsesside või kasutajate poolt
- Kaitsemehhanism peab:
 - Eristama volitatud ja volitamata kasutajaid
 - Võimaldama määrata juurdepääsureeglid
 - Need reeglid kehtestama

Kasutajaliides

- Palju käske, mis operatsioonisüsteemile antakse, on operatsioonisüsteemi endaga seotud juhtimiskäsud:
 - Protsesside loomine
 - I/O operatsioonid
 - Põhimälu haldamine
 - Failisüsteemi manipuleerimine
 - Kaitsemehhanismid
 - Võrguoperatsioonid

Kasutajaliides

- Liides, mille kaudu kasutaja saab süsteemi juhtida
- Võib olla tekstiline käsuintepretaator või graafiline
 - Unixi shell
 - MS-DOSi COMMAND.COM
 - MacOSi GUI+Finder
 - Windows XP GUI+Explorer
 - Windows 8 GUI+start screen

Operatsioonisüsteemi teenused

- Teenused, mida operatsioonisüsteem pakub programmidele
- Programmide täitmine (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 — andmevahetus protsesside vahel (samas või erinevates arvutites), realiseeritakse teadete saatmisega või jagatud mäluga
- Vigade haldus — protsessori, mälu, I/O seadmete ja kasutajaprogrammide vigade avastamine ja neile adekvaatselt reageerimine

Operatsioonisüsteemi lisafunktsioonid

- Mitte kasutajale orienteeritud, vaid süsteemi enda paremaks funktsioneerimiseks
- Ressursside hõivamine ja vabastamine
- Arvepidamine — kes kui palju ja missuguseid ressursse on kasutanud (arvete esitamiseks, statistika jaoks)
- Kaitse ja turvalisus

Süsteemifunktsioonid

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

Süsteemifunktsioonide levinumad tüübid

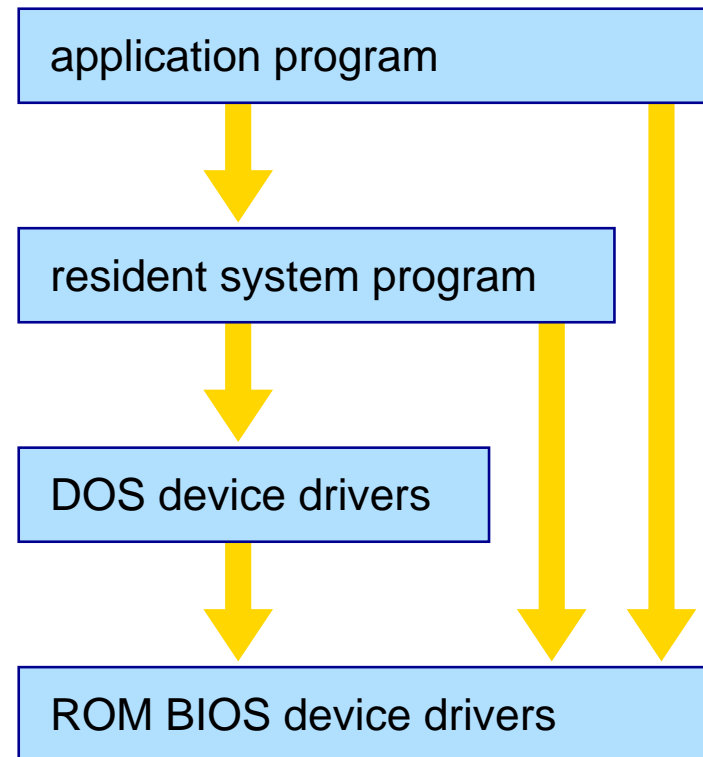
- Protsessihaldus
- Failihaldus
- Seadmete haldus
- Üldinfo
- Side

Süsteemsed programmid

- Aitavad luua mugavat keskkonda rakendusprogrammide jooksumiseks ja arenduseks
- Annavad juurdepääsu süsteemifunktsioonidele ja teenustele kasutajale kättesaadaval kujul
- Kasutaja vaade süsteemile on enamasti defineeritud just süsteemsete programmide kaudu
- Operatsioonisüsteemi tuuma vaatepunktist on tegu harilike rakendusprogrammidega
 - Failide manipuleerimiseks
 - Staatusinfo jagamine
 - ...
- Osa süsteemseid programme on vahel realiseeritud käsuinterepraatori osana

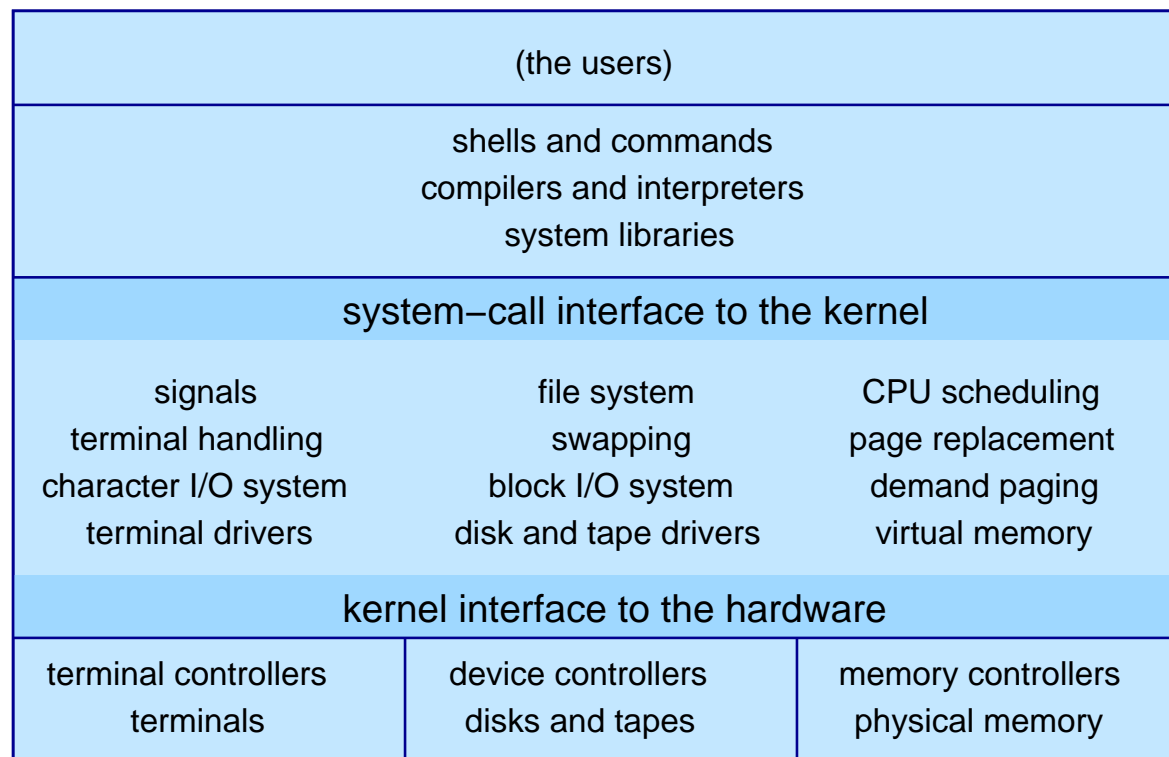
Struktuur: MS-DOS

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



Struktuur: originaalne UNIX

- UNIX oli samuti piiratud riistvara funktsionaalsusega
- Natuke struktuurne, kuid endiselt monoliitne
- Kaks eraldiseisvat osa: süsteemprogrammid ja tuum
- Tuum on kõik, mis jääb allapoole süsteemifunktsioonide liidest



Kihiline lähenemine

- Operatsioonisüsteem 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, QNX, Hurd; natuke ka MacOS ja Windows

Virtuaalmasinad

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

Virtuaalmasinad

- Füüsilise masina ressursse jagatakse virtuaalmasinate loomiseks
- Protsessori(te) jagamine jätab mulje, et kasutajatel on oma protsessorid
- Failisüsteem ja spuuimine (*spooling*) annavad virtuaalsed kettad, printerid ja muud sisend-väljundseadmed
- Harilik ajajaotussüsteemi kasutatav terminal võib olla virtuaalmasina konsooliks
- Jagatud kettad
- Virtuaalsed võrguliidesed virtuaalmasinate vahel suhtlemiseks

Virtualiseerimise tasemed

- Paravirtualiseerimine
 - VM kasutab spetsiaalseid optimeeritud teenuseid igasuguse I/O ning madala taseme pöörduste jaoks
- Füüsilise arvutiga identse liidese loomine
 - Hüperviisor emuleerib kõike I/O seadmeteni välja, VM sees on vastavad seadmedraiverid

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)
- Tänapäeval pakutakse virtuaalmasinatele sageli reaalsest riistvarast erinevat efektiivsemat liidest (*directio*, *enlightened I/O*)

Konteinerid

- Üks operatsioonisüsteemi instants (tuum) pakub rakendusprogrammidele mitut erinevat omavahel isoleeritud töökeskkonda
- Kergekaalulisemad kui päris virtuaalmasinad
- Jagavad keskkonda mitme protsessi vahel
- Piirang: kõigis konteinerites on sama opsüsteem (kasutajataseme programmid võivad erineda, tuum on sama)
- Nõrgem isoleerimine kui virtuaalmasinate puhul: tuum on sama ja sealsed vead saavad teisi konteinereid mõjutada
- Näited: Docker, Linuxi OpenVZ ja LXC, Solarise tsoonid

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
- Praktikas vähe levinud