

ATR0110 ANALOOG- JA DIGITAALTEHNIKA

Kevad 2019

Digitaaltehnika- Arvutid - Operatsioonisüsteemid

Martin Jaanus

NRG-308

martin.jaanus@ttu.ee 56 91 31 93

Õppetöö : <http://isc.ttu.ee>

Õppematerjalid : <http://isc.ttu.ee/martin>

Teemad

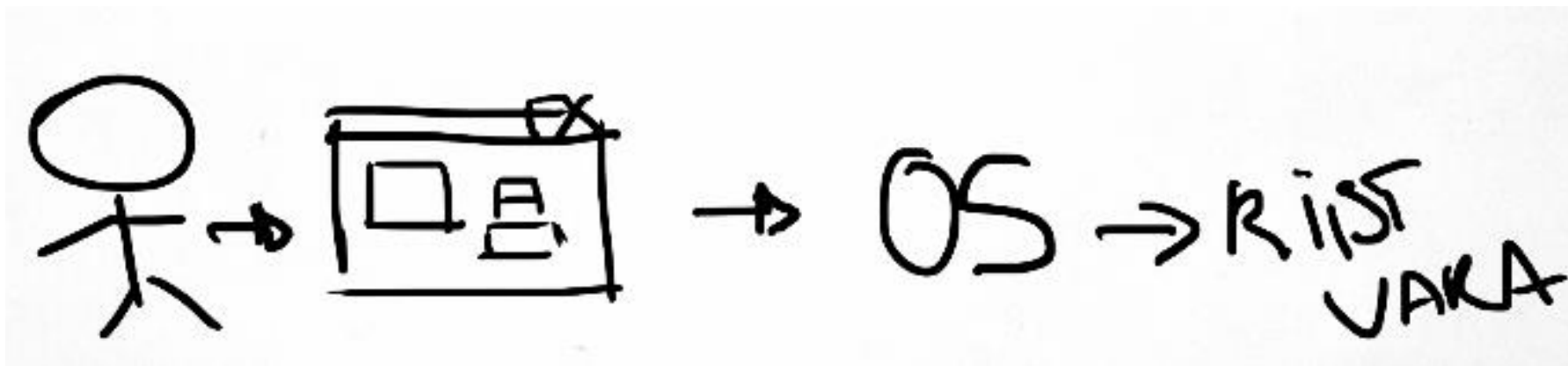
Arvutite töö haldamine (opsüsteemid)

Arvutite muutumine ajas

- Arvutusvõimsus hüppeliselt tõusnud
- Hind drastiliselt langenud
- Miljardid käsud sekundis (kiirus)
- Superarvutid -> triljonid käsud sekundis
- Kaasaskantavad “arvutid” – Interneti leviku mõju

Arvutisüsteemi põhilised komponendid

- Riistvara
- Operatsioonisüsteem
- Kasutaja programmid
- Kasutaja



Mis on operatsioonisüsteem

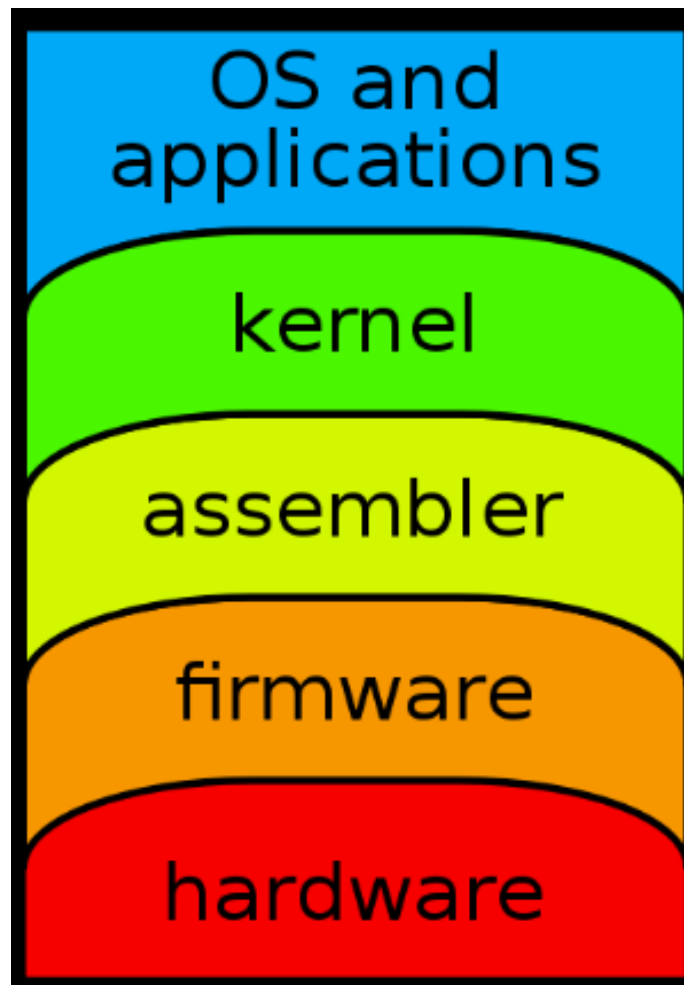
60-70 ndatel – tarkvara, mis kontrollib riistvara.

Tänapäeval:

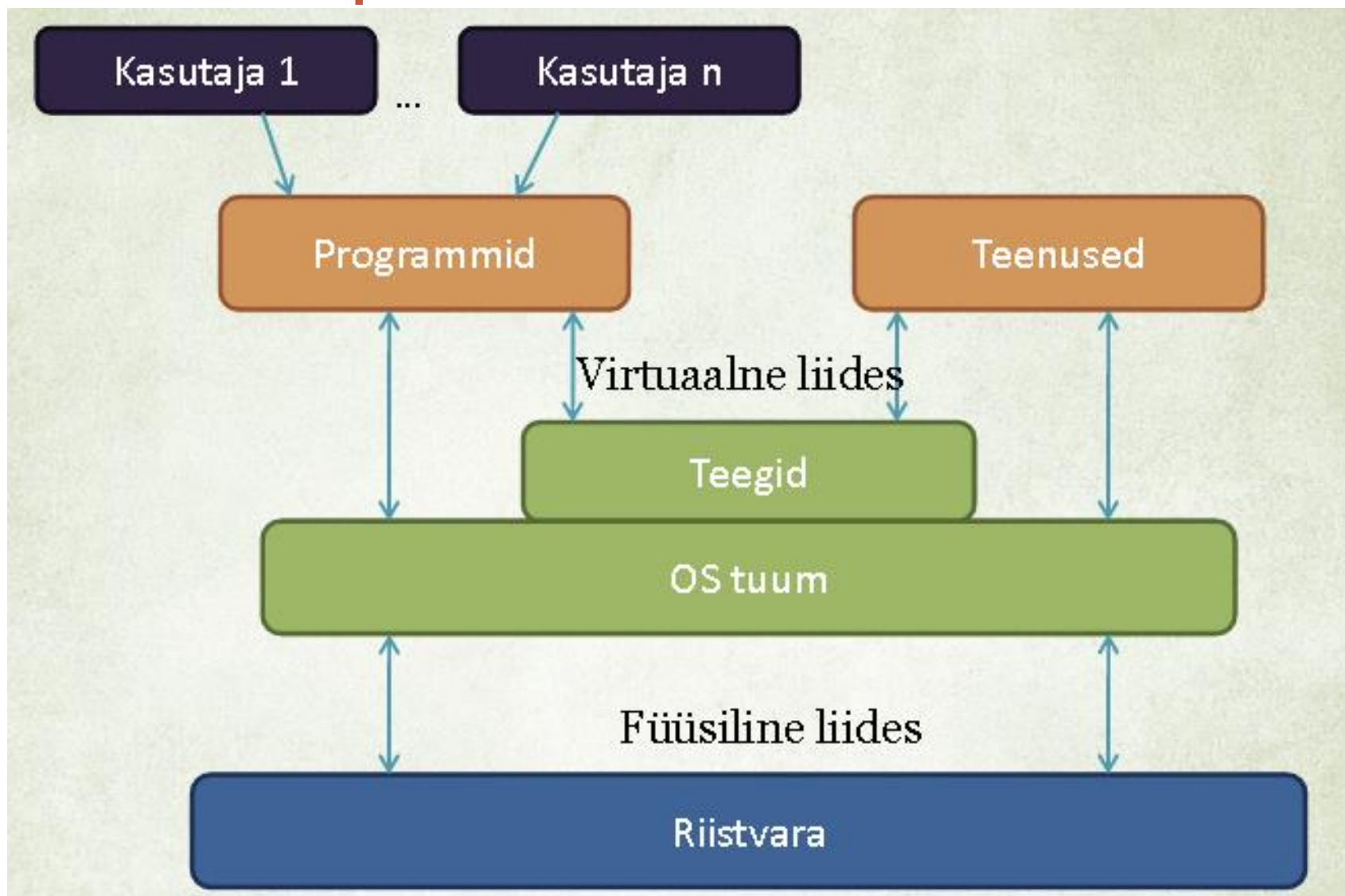
- programm, mis käitub kui vahendaja kasutaja ja riistvara vahel
- Kasutajaprogrammid saavad samal ajal töötada
 - Kui ei ole ettevaatlik, siis võivad mõjutada üksteise tööd ja tulemusi
- Operatsioonisüsteem on kui vahekiht, mis eraldab programme riistavarast ja pakub teenust, mille kaudu saavad programmid koos töötada ohutult ja efektiivselt.

Mis on operatsioonisüsteem

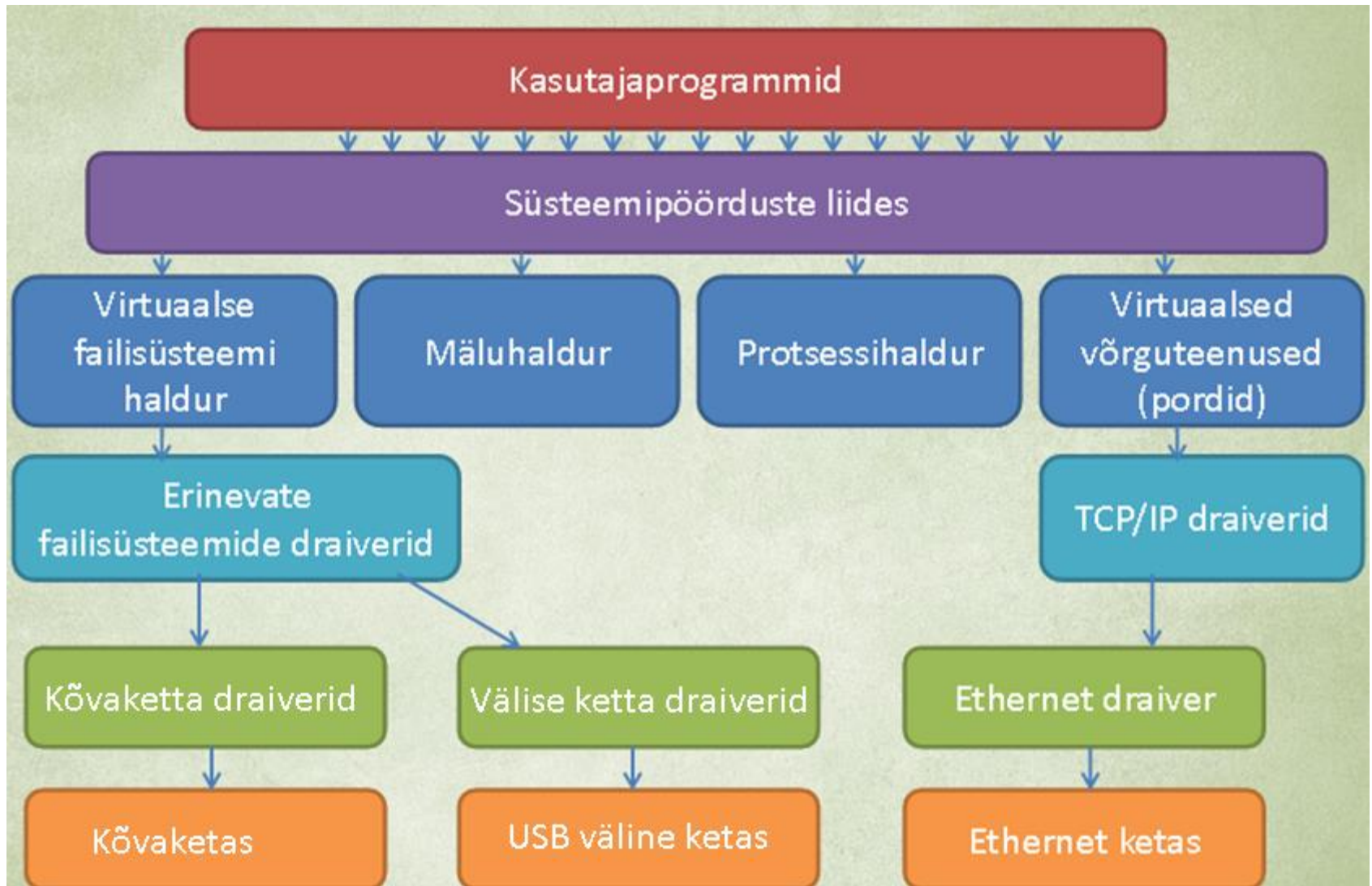
- Varustab kasutajat vahendite komplektiga, mille abil on võimalik programme käivitada
- Tarkvara põhiosa OS-is: kernel(tuum)
- Tuum -Sild rakendusprogrammide ja tegeliku andmete töötlemise vahel riistvara tasemel
- OS on resursside haldaja
- Protsessid, mälu, I/O



Mis on operatsioonisüsteem



Operatsioonisüsteemi tuum



Operatsioonisüsteemi eesmärgid

- **Riistvara varjamine läbi abstraktsioonide**
- Abstraktsioon - tarkvara, mis varjab madalama taseme detaile, OS muudab reaalse, füüsiline (seaded, mälu, aeg) maailma virtuaalseks

Miks on abstraktsioone vaja?

- Kood välisseadmete kontrolliks ei ole standarteetritud -> **draiverid**
- OS annab uue taseme funktsionaalsusi läbi abstraktsioonide. (nt. programmid tegelevad failidega mitte kettaga)
- OS muudab riistvara mitmeks virtuaalseks masinaks.
- Iga protsess näeb riistvara läbi abstraktsiooni
- OS saab lisada turvalisust läbi abstraktsioonide
- Resursside määramine protsessidele
- Pakub sõbralikku kasutajaliidest (Kasutaja „näeb ja tunneb“ os läbi kasutajaliidese)

Opsüsteemi ajaloo

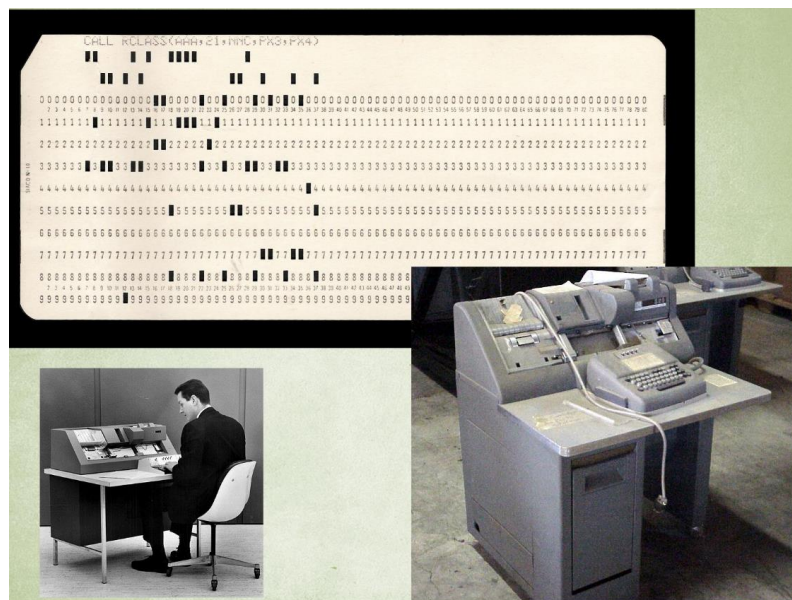
- 1940-1950 . Arvutid ei sisaldanud opsüsteeme , üks programm, masinkoodis (assembleris), peimitiivsed
- Esimene OS :General Motors Research Laboratories1950-tel IBM 701 arvutile (1 töö korraga, töötas tunde, päevi ilma kasutaja sekkumiseta)
- Suured masinad
- Käivitati konsoolist
- Sisend :perfokaart või -lint
- Väljund : printer või perforaator
- Protsessori kasutus ebaefektiivne



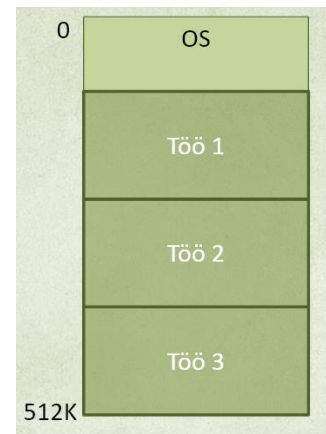
IBM 701 , esimene kommertslik arvuti (1952)
wikipedia

Opsüsteemi ajaloost

- 1960 – juba efektiivsemad
- Mitu tööd korraga (üks protsess kasutab protsessoriot, teine I/O)
- Seadmed – perfokaardid, magnetlindid, magnetkettad, tänapäevamõistes kuvar ja klaviatuur.
- Optimeerimine tööde vahel, draiverid



Mälupilt →



Arvuti töö 60ndatel

- Programmi tekst perfolindile/kaardile
- Magnetlint kompilaatoriga seadmesse ja seade ühendati (mount)
- Kaardilugeja lugemine kaardid -> sisu magnetlindile
- Kompilaatori käivitamine ->assembleri kood
- Linkur teiselt magnetlindilt -> laademoodul
- Laademoodul mällu ja selle käivitus

Arvuti töö 60ndatel

- Juba mimekasutaja süsteem
- Virtuaalmälu (suuremad programmid) – võimaldas kallist operatiivmälu asendada odavama ketta või lindimäluga. Ebavajalikud andmed kirjutati välismällu.
- CPU kasutus efektiivsem.

1970

- Multiprogrammsed süsteemid –“portsjonites” töötlemine
- Ajajaotus- realajas töötavad rakendused
- Personaalarvutite tekkimine tänu mikroprotsessorite arengule– Apple II(1977)
- Odavamad, väiksemad aga ei toetanud : Mitut kasutajat, multiprogrammist, failisüsteemi kaitset.
- Arvutivõrkude kiire areng

Kaks arengusuunda

- Unix
- Microsoft (DOS, 80ndatel Windows)

http://www.youtube.com/watch?v=2_iNKWO3RXY



1980

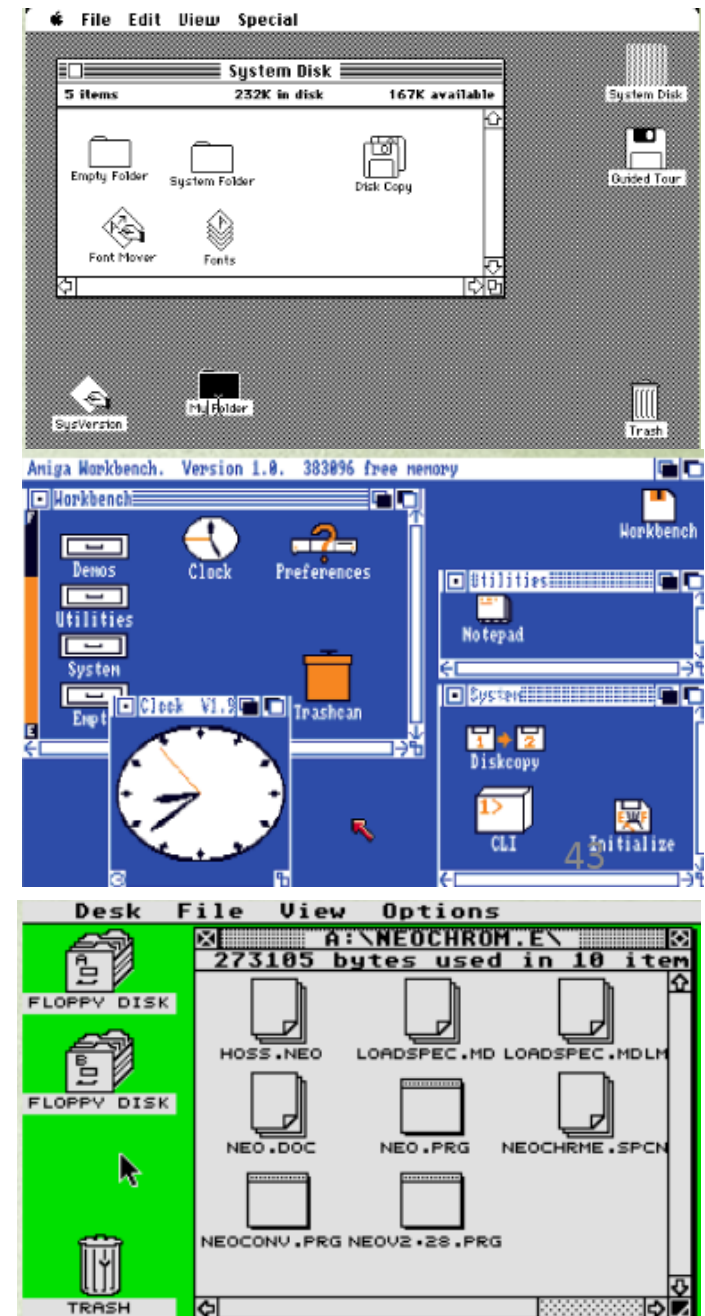
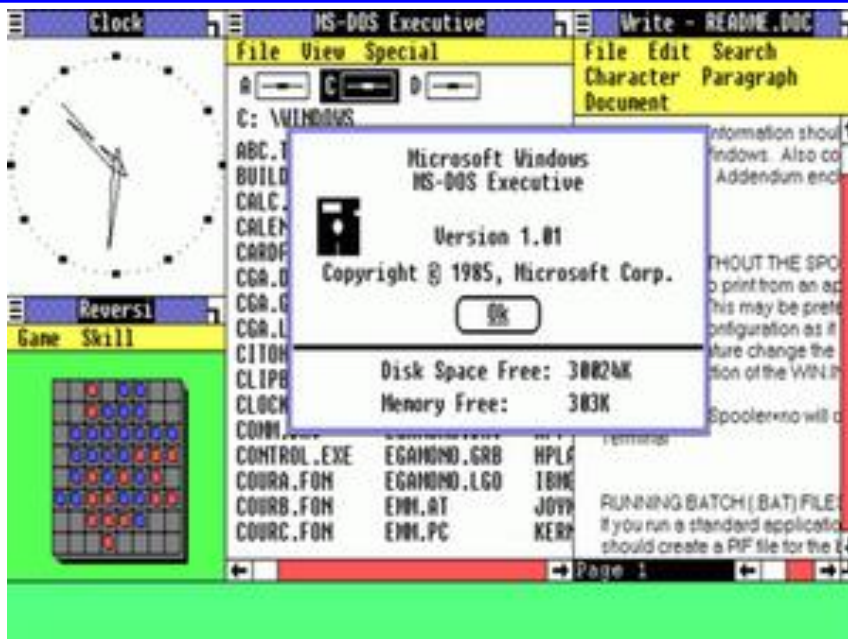
- Personaalarvutite kiire areng – tänaseni.
- Esimene PC – IBM PC (1981)
- Universaalsed, tekkisid „valmis“ tarkvarapaketid“
- Tekkis GUI (graafililine kasutajaliides)



Pilt- wikipedia

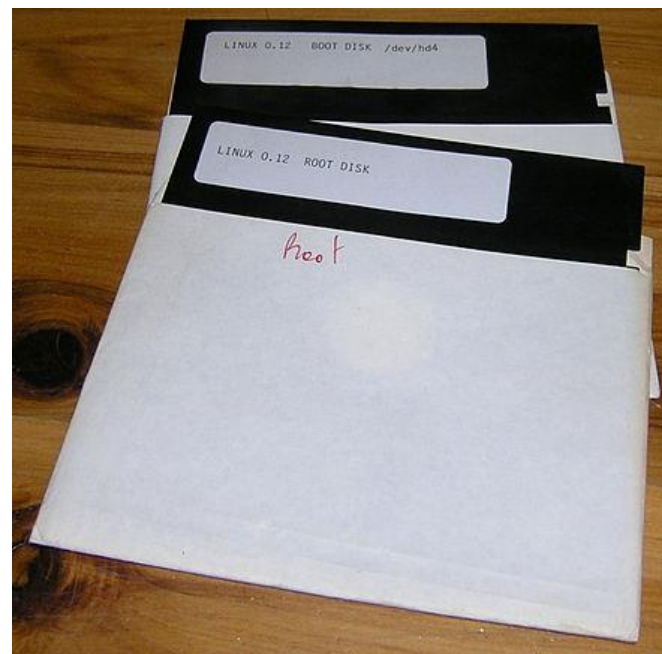
GUI

- 1984 Apple Macintosh – esimene edukas
- 1985 Atair - Graphical Environment manager
- 1985 Commodore workbench
- 1985 MS Windows
- <https://www.businessapac.com/history-of-windows-from-ms-dos-windows-1-0-to-windows-10/>



1990

- Kasutajasõbralikus
- “Plug-and-play” – Võimalus lisada/eemaldada riistvara komponente ilma OS-i ümber konfigureerimata
- Kasutaja profiilid – Isikupärastamine – paroolid
- Linux (1991), Open source , GPL
- WWW (1993)
- Suureneb andmete turvalisuse probleem



Linux'i algvariandi kettad (wikipedia)

2000 - tänapäev

- Vahevara (middleware)
- Veebiteenused
- Online kaubandus
- Järjestikune programmeerimine -> Paralleelne programmeerimine
- Äri ja kodu OS ühendamine (näiteks Windows XP ja edasi) – varem olid eraldi seeriad.
- Hajussüsteemid („arvuti“ moodustub paljudest arvutitest)
-
- Vahetvara-Tarkvara, mis seob kokku kaht muidu iseseisvat rakendusprogrammi. Näiteks on olemas rida vahetarkvara tooteid, mis ühendavad andmebaasi veebiserveriga. See lubab kasutajal nõuda andmebaasist andmeid veebibrauseri poolt kuvatava blanketi täitmise teel ja veebiserveril saata tagasi dünaamilisi veebilehekülgi, mis vastavad kasutaja profiilile ja nõudmistele.

2000- tänapäev

ManusOS (embedded) . Nutiseadmed

- Eesmärgid veidi teised.
- Vähe(m) mälu, kood lihtsam
- Kiirus - vs aku eluiga, puuteekraan, väike ekraan.
- Kasutusel sageli vaid „terminalina“ Enamus arvutusi toimub serveris

--Missioonikriitilised (nt lennujuhtimissüsteemid)

--Ärikriitilised (suured veebiteenused, serverid)

Virtuaalmasinad (tarkvara abstraktsioon – juhib riistvara, mida tegelikult ei ole , palju OS korraga samas masinas)

Tänapäev ja tulevik (?)

- Lihtsus, kasutajasõbralikus.
- Online süsteemid
- Ei failisüsteemidele !
- OS internetis – arvuti vaid terminal (Chrome OS).
- Tehisintellekt
- Andmekorilus (suured andmed, big data, data mining)
- Vajalik väga kiire võrguühendus

OS-i ülesanded

- Protsesside haldamine(otsustab millal ja milline protsess kui palju CPU-d saab)
- Põhimälu haldamine(millal ja kuidas mälu antakse)
- Failide haldamine
- I/O haldamine
- Sekundaarmälu haldamine
- Võrgu haldamine
- Kaitsesüsteem
- Käsuinterpretaator

Protsesside haldamine

- Ühest programmist -> mitu protsessi
- Igal oma olek ja erinev käsuloendur
- Süsteem koosneb protsesside komplektist
- Igal ühel “oma” CPU kasutada

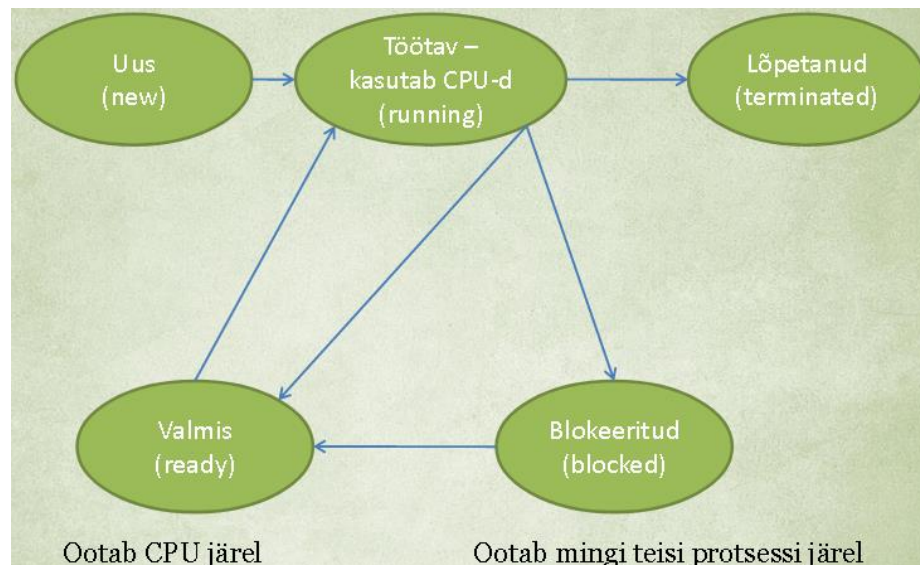
Protsessijuhtimine:

- Protsesside loomine ja kustutamine
- Protsesside peatamine ja taastamine
- Protsesside sünkroniseerimine
- Protsesside omavahelise kommunikatsiooni organiseerimine

Protsesside haldamine

Protsessori aja andmine

- reaalselt saab ühel protsessoril käia üks protsess korraga
- Protsessi olekud:
 - Jookseb (running) täidetakse protsessoris
 - Valmis (ready) –saaks täita kui oleks protsessor vaba
 - Blokeeritud (blocked) –ootab mingit kindlat sündmust



Protsesside haldamine

- Kasutaja jooksub programmi -> luuakse protsess -> protsess pannakse readylisti (new)
- Protsess jõuab listi algusesse -> antakse protsessori aega (running)
- Dispetšer (OS-i “komponent”, mis annab CPU readylisti alguses olevale protsessile)
- “Ärkvel olevad” protsessid (need, mille staatus on „ready“ või „running“)
- **Ükski protsess ei tohi protsessorit täielikult endale saada**
 - Katkestuskell -> katkestus
 - Protsessor võetakse ära (ready)

Kui protsess tahab I/O tegevust, siis loovutab vabatahtlikult protsessori (blocked)–Peab I/O järel ootama

Protsesside haldamine

- Kui süsteem saab otsustada millist protsessi järgmisena täita -> siis peab tal olema mingi strateegia selleks.

Protsessi haldus teatud eesmärgiga nt:

- Max nr protsesse saab joosta teatud aja jooksul
- Min aeg, mida protsess ootama peab
- Ära hoida igavest ootamist
-

Protsesside haldamine

First Come, First Served (FCFS)

Kes ees, see mees.

- Lihtsaim
- Mitteennetav (ei vahetata välja)
- Protsess, mis esimesena nõuab CPU-d, teenindatakse esimesena
- Kõik saavad teenindatud, lühikesed peavad kaua ootama.
- Ei ole kasulik interaktiivsetele protsessidele
- Tänapäeva harva kasutuses

Protsesside haldamine

Shortest Job First (SJF)

- Lühim töö järgmisena
- Ooteaeg väheneb
- Mitteennetav (ei vahetata välja)
- Interaktiivsed protsessid on lühikesed
- **Peab teadma kaua protsess jookseb -> tihti seda infot pole.**

Protsesside haldamine

Shortest remaining time First (SRTF)

- Lühim järgi jäänud aeg järgmisena
- Ooteaeg väheneb
- Ennetav (vahetatakse välja, kui tuleb lühen kokureeriv protsess)
- Enamasti efektiivsem ,kuid.. Protsess kohe lõpetamas, vahetatakse välja.. Kogu aeg vahetatakse–Ajakulu vahetamiseks on suurem kui olemasoleval lõpuni joosta
- **Peab teadma kaua protsess jookseb -> tihti seda infot pole.**

Protsesside haldamine

Round Robin

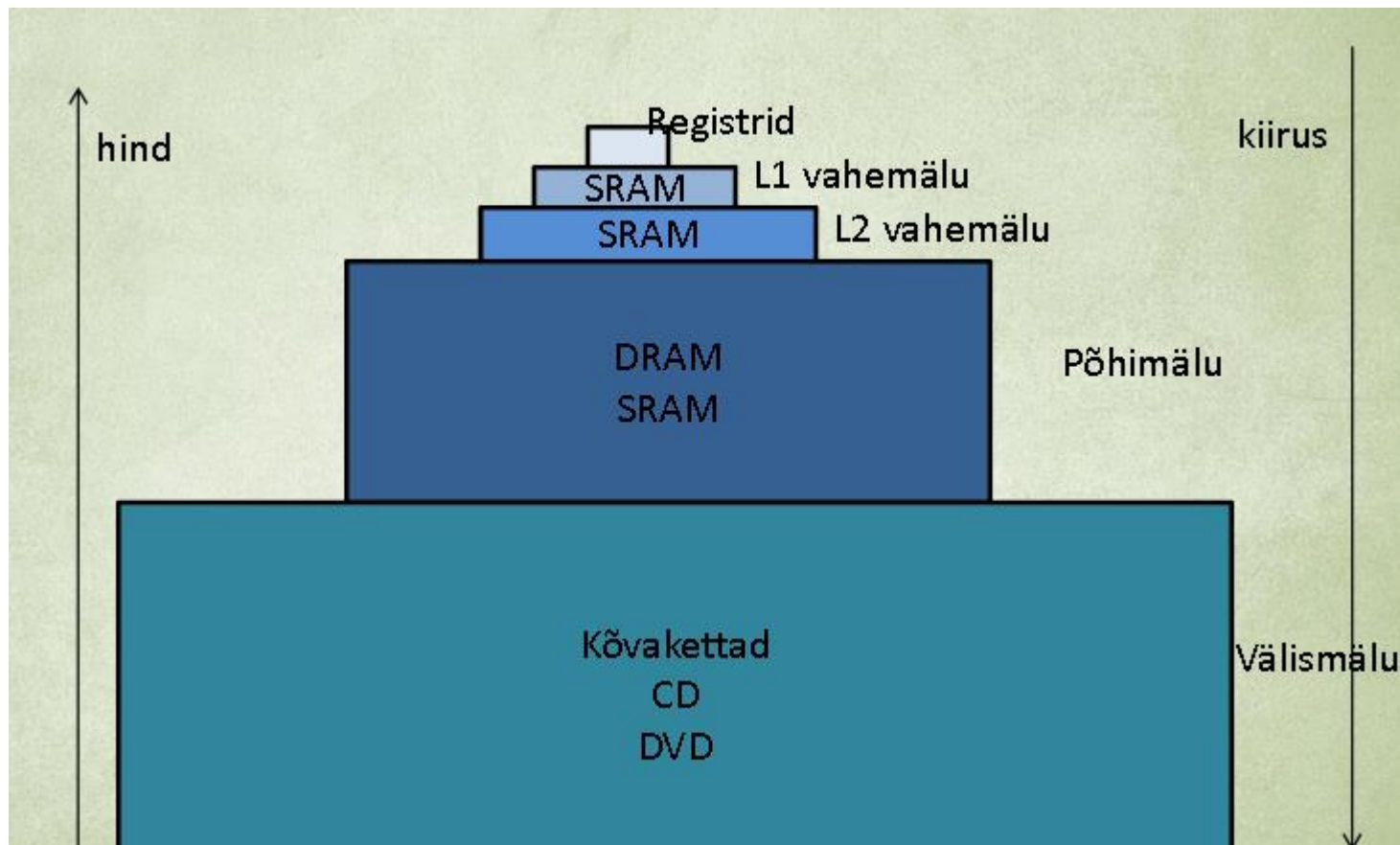
- Ajajaotussüsteemides
- Protsessori aega jagatakse kordamööda konstantsete taktidena
- Ei ole parim ooteaeg aga parim reaktsiooniaeg
- Tähtis takti valik
- – Kui väike protsesside kogukestvusega, siis on ümberlülituste lisakulu suur
- –Liiga suur takt taandab FCFS algoritmiks

Mälu haldamine

- Üks protsess mälus?
- Kui mitu, kas kõik saavad võrdse osa?
- Kas jagada mälu partitsioonidesse?
- Kas protsesside mälupiirkond peab tulema järjest

- OSi komponent, mis tegeleb süsteemi mälu organiseerimisega ja mälu haldamise strateegiatega
Kuidas olemasolevat mälu protsessidele anda (kui saadaval) suhtleb mälu haldamise riistvaraga.

Mälu haldamine



Mälu haldamine

- Strateegiad –Sissetoomis (fetch) (otsustab millal tuua uus programm või andmed sekundaarsest mälust põhimällu)
- Nõude sissetoomine (demandfetch) - kui andmeid vaja, siis pannakse need alles põhimällu.
- Ennetav sissetoomine (anticipatoryfetch) (üritame ette aimata, mida vaja läheb ja toome need enne põhimällu kui nende poole pöördutakse. Võidakse tuua ka asju, mida ei lähe vaja – kasutu töö)
- Asetamise (placement) - kuhu mälus panna)
- First fit
- Best fit
- Worst fit
- Asendamise (replacement) (kui mälu täis, tuleb otsustada, mida asendada)

Mälu haldamine

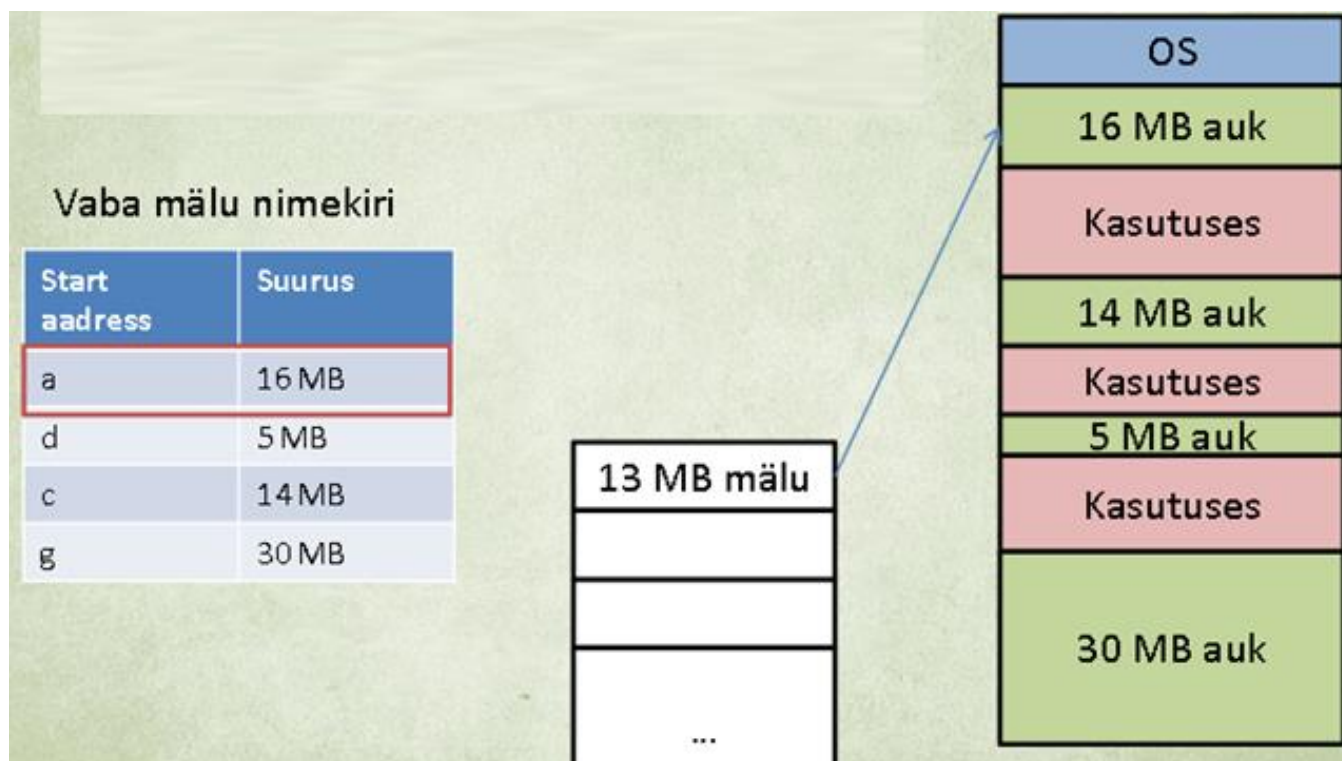
- Mälu fragmenteerimine
- On piisavalt mälu aga seda ei ole võimalik kasutada
- Liigid–Välimine (mälu on, kuid see ei ole pidev)
- Sisemine (kinnistatud mälu on suurem kui tellitud. Vahel ei ole vajalik aga seda ei saa kasutada)
- Väline fragmenteerumine on omane dünaamilise mälujaotusele

- Erinevad strateegiad fragmenteerumise minimeerimiseks.

Mälu haldamine

First Fit (FF)

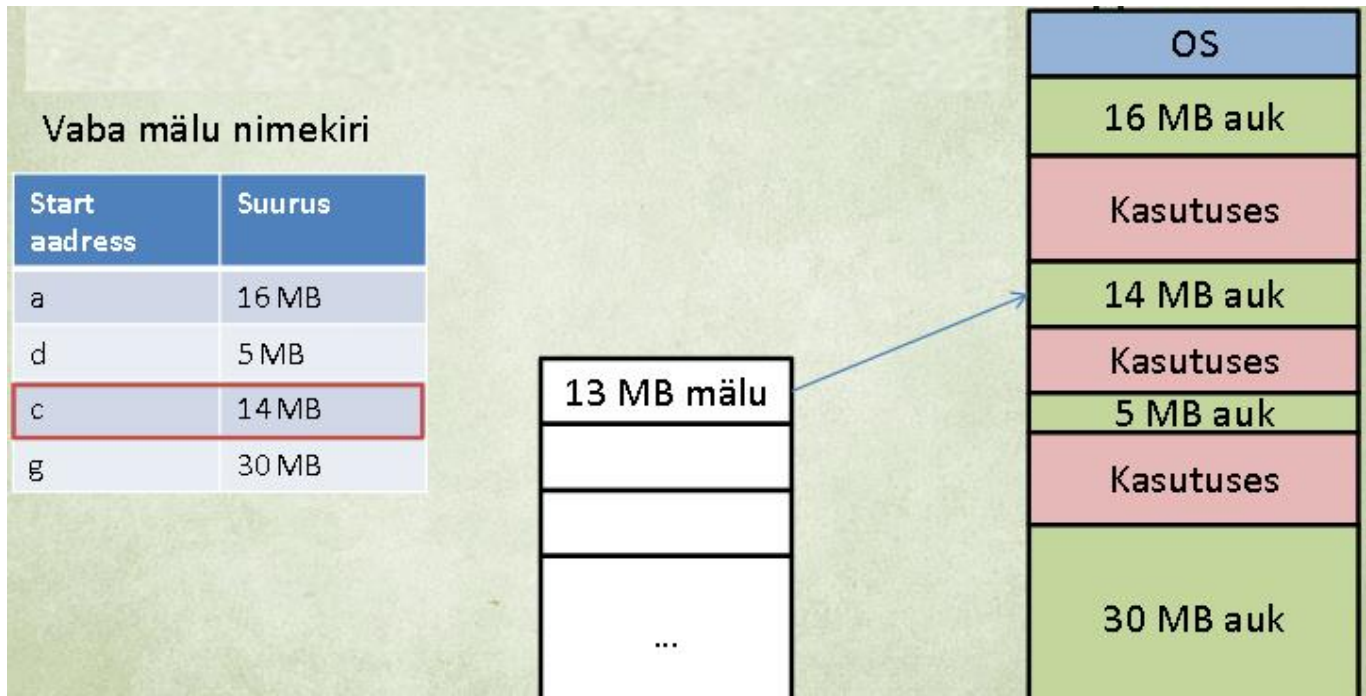
- Töö pannakse põhimälus esimesse piisavalt suurde vabaolevasse auku. Otsuseid tehakse kiirelt.



Mälu haldamine

Best fit

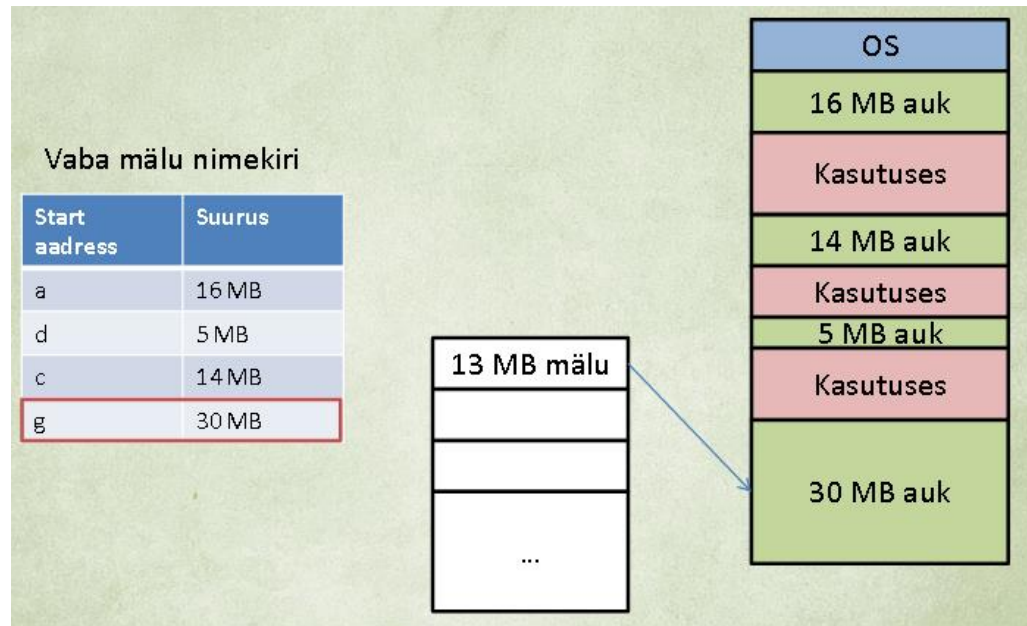
- Süsteem paneb sissetuleva töö peamälu auku, kuhu see kõige väiksemate kadudega mahub –ehk üritab saavutada parimat pakkimist. **Võtab aega -> parima augu leidmine**



Mälu haldamine

Worst Fit

- Paneb töö põhimälus kohta, kus kõige suuremad kaod (ehk kõige rohkem ruumi jääb üle)
- Kõlab imelikult, eks ju? Aga -> paneme töö suurde auku. See, mis alles jääb on ka suur ja saame sinna uue töö panna.



Mälu haldamine

- FF ja BF paremad kui WF.
- FF-võib põhjustada välimise fragmenteerumise
- Ülesannete näited
- <http://isc.ttu.ee/kats/OS6.pdf> alates LK 75