



LUT
University

CT30A3370 - KÄYTTÖJÄRJESTELMÄT JA SYSTEMIOHJELMOINTI 6 OP

Jussi Kasurinen (etu.suku@lut.fi)

Osa kalvoista © Timo Hynninen.

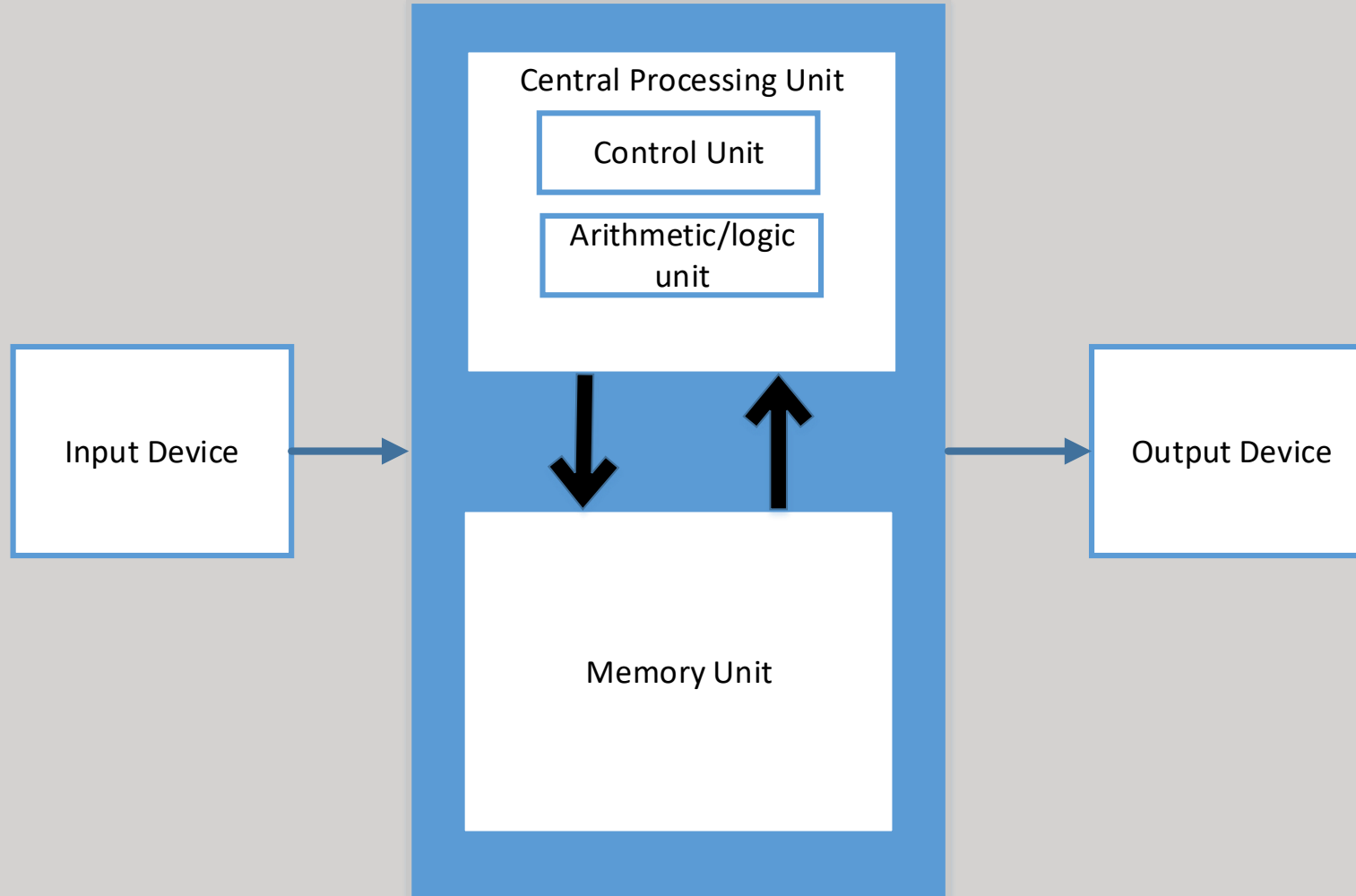


EDELLISELLÄ LUENNOLLA: KÄYTTÖJÄRJESTELMÄ JA TIETOKONEARKKITEHTUURI

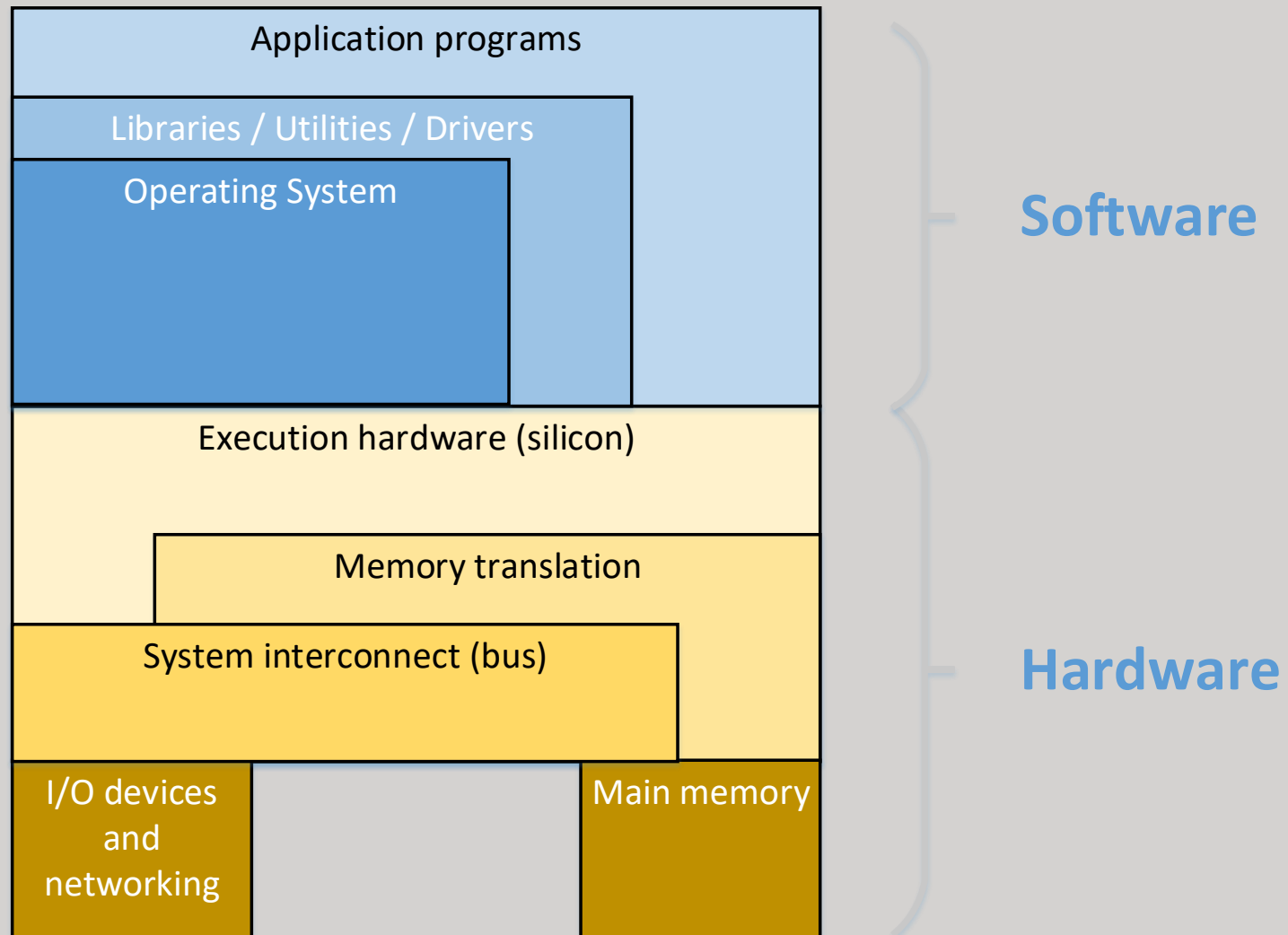
CT30A3370 - Käyttöjärjestelmät ja
systemiohjelmointi



RE: VON NEUMANN -ARKKITEHTUURI



MODERNI TIETOKONE



VRT. ANDROID (GITHUB DEVACADEMY)

4

APPLICATIONS

Home	Dialer	SMS/MMS	IM	Browser	Camera	Alarm	Calculator
Contacts	Voice Dial	Email	Calendar	Media Player	Photo Album	Clock	...

3

APPLICATION FRAMEWORK

Activity Manager	Window Manager	Content Providers	View System	Notification Manager
Package Manager	Telephony Manager	Resource Manager	Location Manager	...

2

LIBRARIES

Surface Manager	Media Framework	SQLite	WebKit	Libc
OpenGL ES	Audio Manager	FreeType	SSL	...

ANDROID RUNTIME

Core Libraries
Dalvik Virtual Machine

1

HARDWARE ABSTRACTION Layer

Graphics	Audio	Camera	Bluetooth	GPS	Radio (RIL)	WiFi	...
----------	-------	--------	-----------	-----	-------------	------	-----

LINUX KERNEL

Display Driver	Camera Driver	Bluetooth Driver	Shared Memory Driver	Binder (IPC) Driver
USB Driver	Keypad Driver	WiFi Driver	Audio Drivers	Power Management

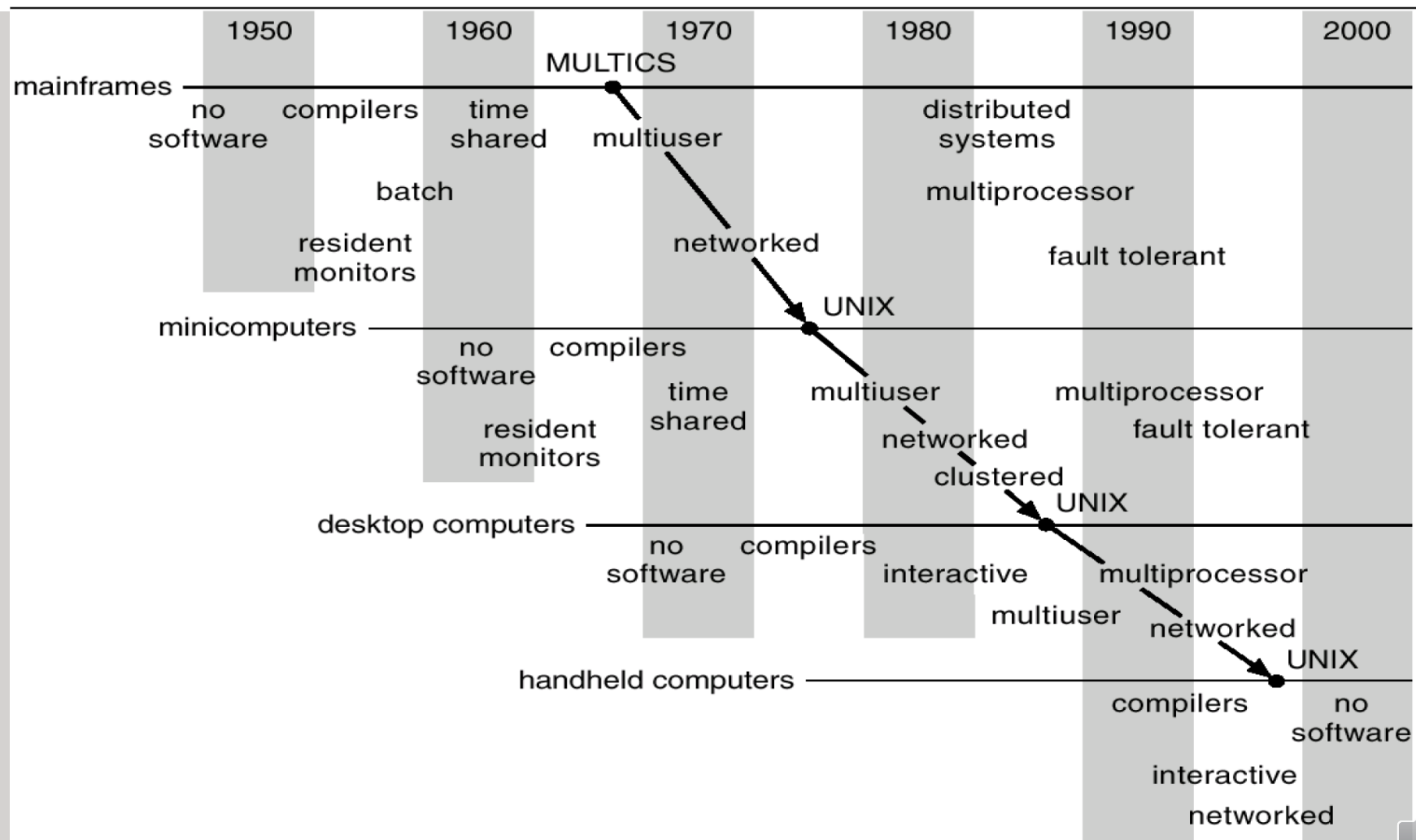


KÄYTTÖJÄRJESTELMÄ YLEISESTI

- Käyttöjärjestelmä käytännössä hoitaa tiedonvaihtoa ohjelmisto- ja ajuritason kanssa (ja avulla) raudan suuntaan.
- Käyttöjärjestelmä mahdollistaa samanaikaisen ohjelmistojen ajamisen, resurssien hyödyntämisen sekä toiminnan valvonnan.
 - Prosessit, säikeet ja niiden valvonta
 - Levynhallinta
 - Suoritinajan jakaminen



VRT: NYKYISET KÄYTTÖJÄRJESTELMÄT, KEHITYS



KÄYTTÖJÄRJESTELMÄN RAKENNE JA OMINAISUUDET

CT30A3370 - Käyttöjärjestelmät ja
systemiohjelmointi



VANHAT KÄYTTÖJÄRJESTELMÄT

- Alkujaan koneet oli ihmisiä kalliimpia
 - Kun tietokoneet oli oikeasti kallista tavaraa, oli järkevää pyrkiä optimoimaan laitteiston toimintaa
- Ensimmäisessä käyttöjärjestelmien kehitysvaiheessa meillä oli käyttäjät, jotka yksi kerrallaan olivat päätteen avulla yhteydessä tietokoneeseen
 - Käyttöjärjestelmä oli lähinnä ohjelmistokirjasto, joka tarjosi paljon hyödyllisiä aliohjelmia



VANHAT KÄYTTÖJÄRJESTELMÄT

■ Nk. Eräajojärjestelmiä

- Lataa ohjelma, aja ohjelma, printtaa ohjelman tulos
- Eräajossa ei kuitenkaan ollut / ole mitään suojamekanismeja:
 - Mitä jos ajettavassa ohjelmassa on vaarantava virhe, joka kaataa koko tietokoneen?
- Lisäksi, Syöttö ja tulostus sekä suorittimen laskenta olivat rinnasteisia toimintoja
 - Ohjelmalla on suora pääsy muistilaitteelle
- Jossain vaiheessa keksittiin muistilaitteiden ohjaimet
 - Käyttöjärjestelmälle laitetaan pyyntö I/O-operaatiosta, käyttöjärjestelmä vaihtaa prosessia, ja jatkaa ohjelman suoritusta sen jälkeen. Laiteohjain lähettää keskeytyksen, kun I/O-toiminto on valmis ja edellinen prosessi voi jatkaa hommiaan!

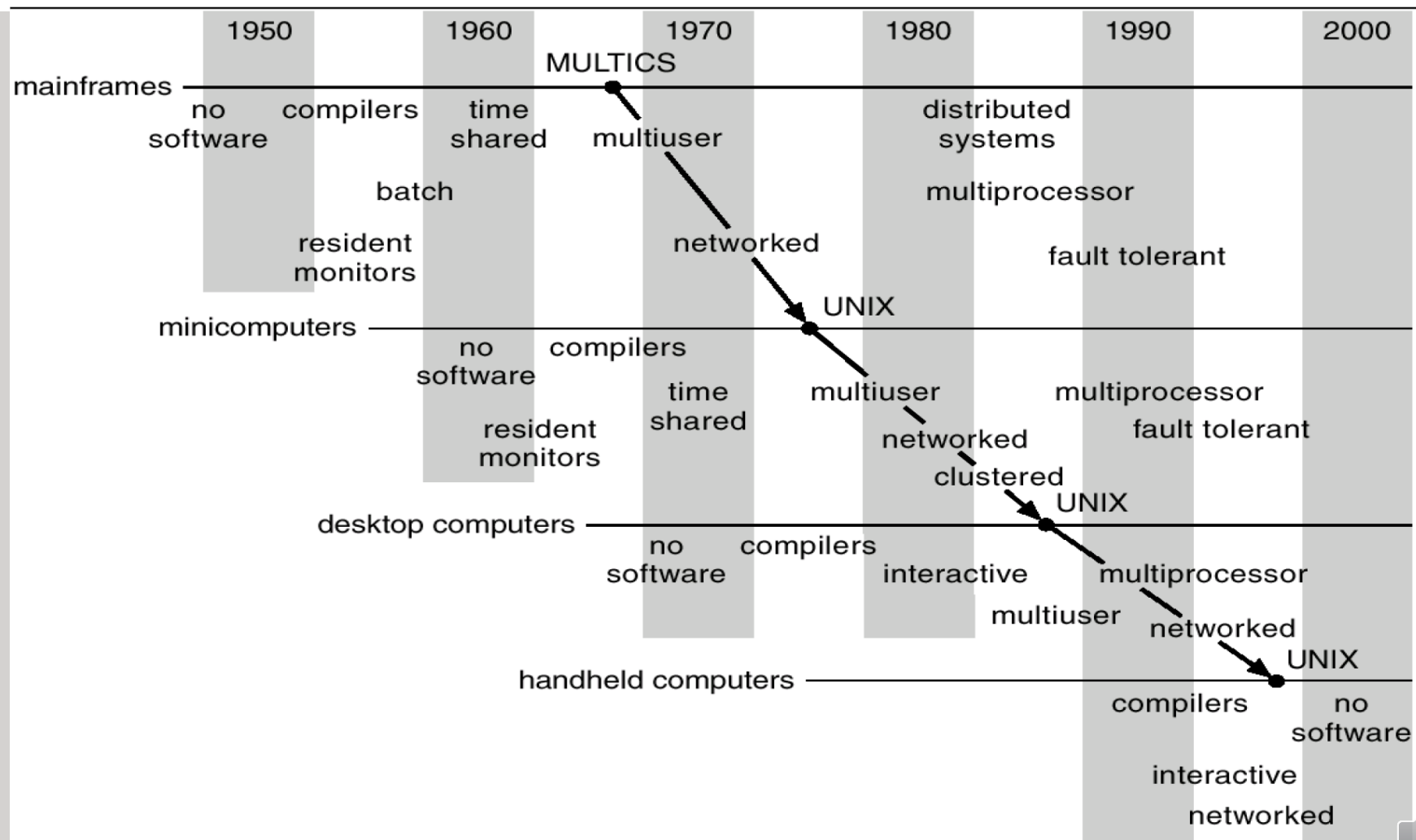


VANHAT KÄYTTÖJÄRJESTELMÄT

- **Sittemmin: Muistinhallinta ja siirrettävä ohjelmakoodi**
 - Moniajon käsite: meillä voi olla useita ohjelmia, joita suoritetaan samaan aikaan
 - Meillä voi olla useita käyttäjiä, jotka käyttävät järjestelmää samanaikaisesti
 - Pienet ohjelmat eivät tällöin joudu odottamaan isojen ollessa tiellä
 - Eriyttää syöttö- ja tulostusoperaatiot suorittimen laskennasta.
- **Kuitenkin moniajo vaatii jonkinlaista muistinhallintaa**
 - Yksi ohjelma ei saa kaataa toista tai kaataa koko konetta!
 - Huono ominaisuus: käyttöjärjestelmän pitäisi pitää kaikista ajettavista ohjelmista, sekä kaikkien ohjelmien vaikutuksista toisiinsa kirjaa!



VRT: NYKYISET KÄYTTÖJÄRJESTELMÄT, KEHITYS



VANHAT KÄYTTÖJÄRJESTELMÄT

- Jossain kohtaa tietokonelaitteiden hinnat alkoivat halveta
 - Koneista tehokkaampia \Rightarrow useat käyttäjät pystyivät käyttämään yhtä tehokasta mainframea tyhmien päätteiden kautta.
 - Suoritinaikaa jaettiin useiden käyttäjien kesken \Rightarrow eli suoritinaikaa uhrattiin paremman vasteajan saamiseksi käyttäjille!
 - Mitä enemmän käyttäjiä, sen hitaammaksi järjestelmät tuli

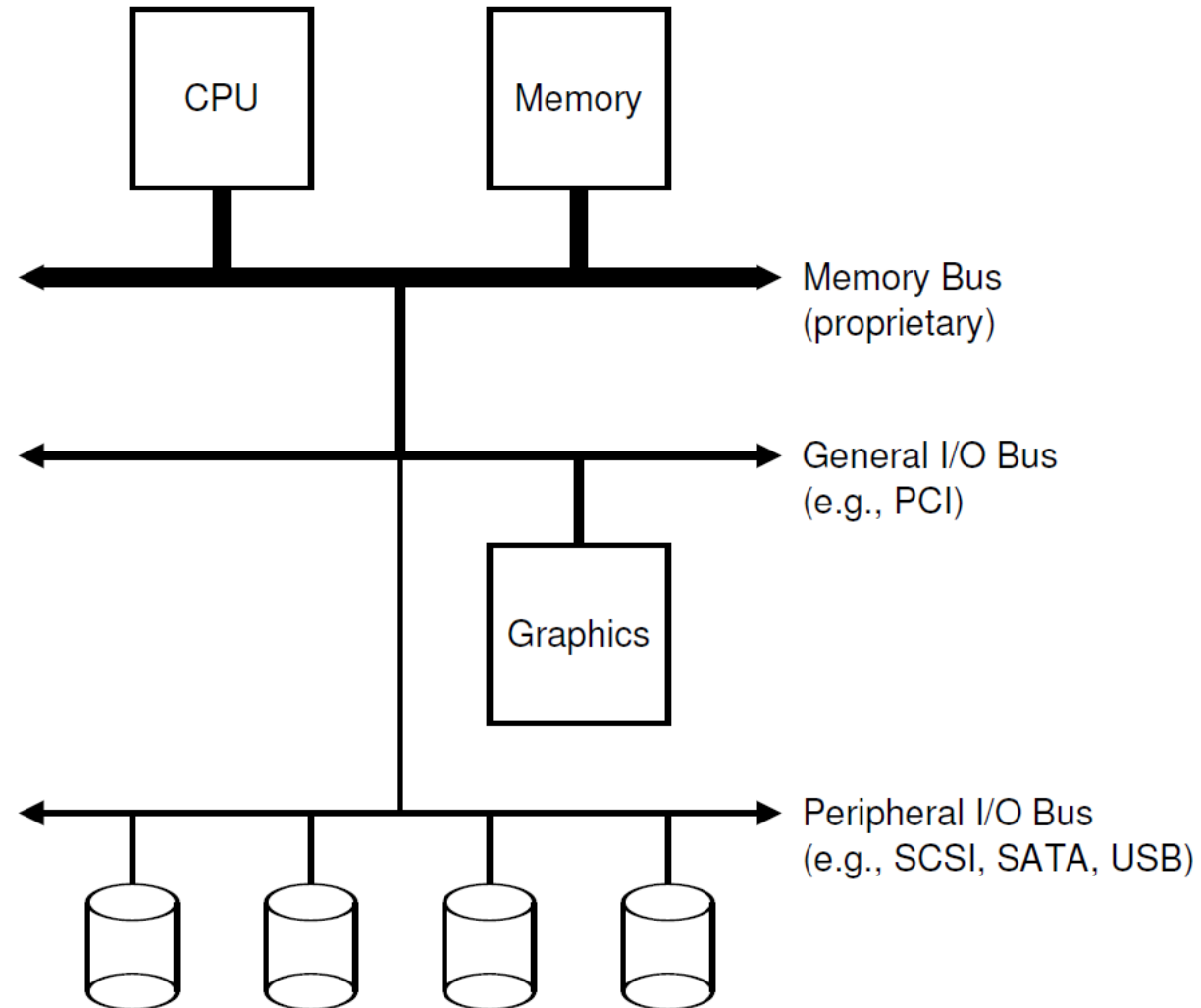


NYKYISET KÄYTTÖJÄRJESTELMÄT

- Lopuksi tietokonelaitteista tuli niin halpoja, että jokaiselle työpöydälle oli varaa hankkia sellainen
 - Yhdellä käyttäjällä yksi tietokone
 - Aluksi poisti ongelmia usean käyttäjän järjestelmiin verrattuna, nykyisellään muistinhallinta ja sivistyneemmät moniajomahdollisuudet tekivät uudelleen paluun
 - Esimerkiksi, PC-koneiden yleistyessä MS/DOS tai Windows 95 olivat kategorisesti vain yhden käyttäjän järjestelmiä
- Verkottuminen (Internet, ethernet) mahdollisti resurssien (levytila, suoritinaika) jakamisen tietokoneiden välillä helposti.

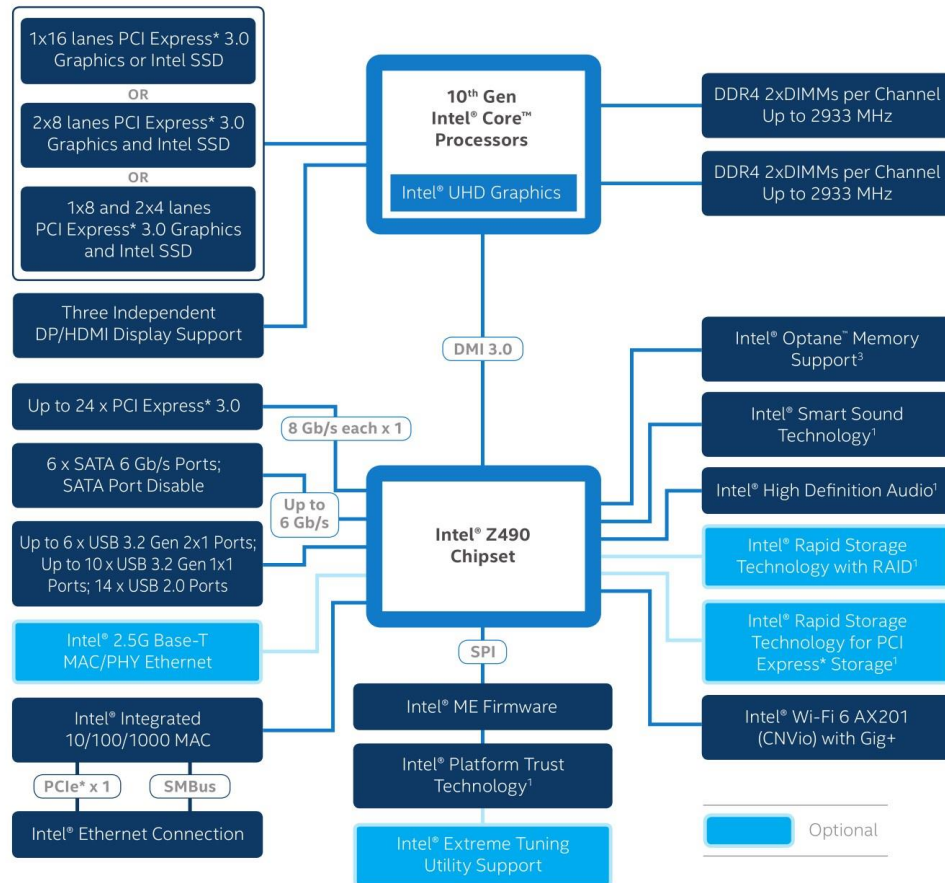


NYKYINEN (PC) TIETOKONE PERIAATTEELLISELLA TASOLLA

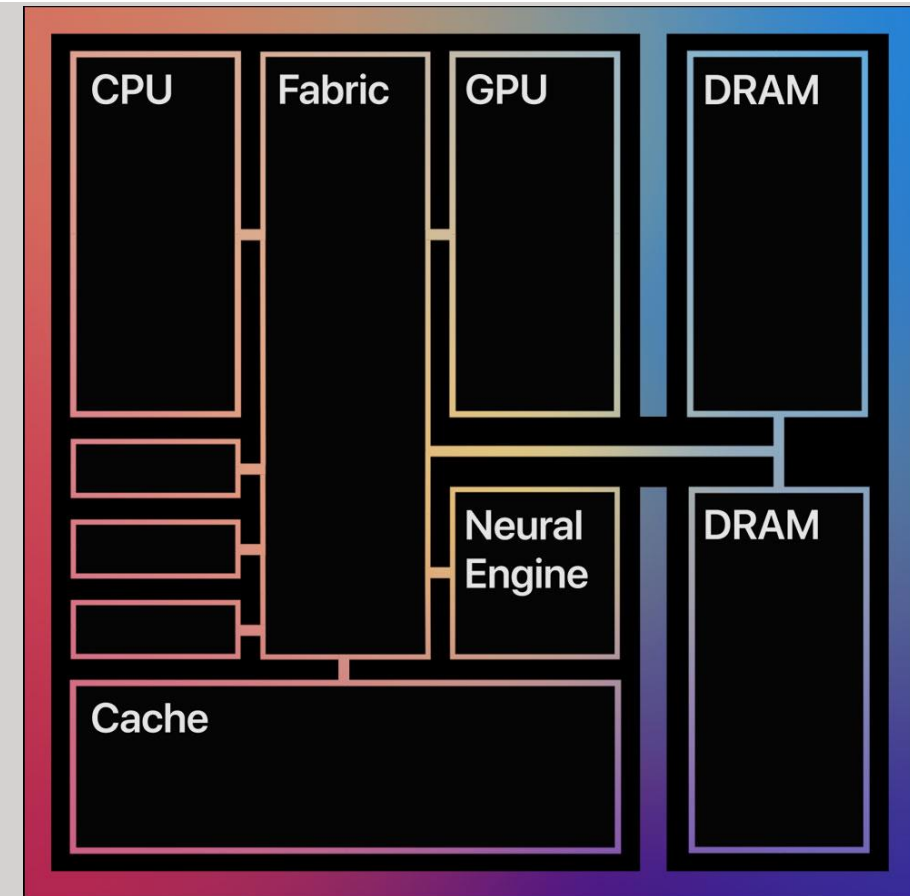


PARI NYKYISTÄ PROSESSORI/KONEARKKITEHTUURIA

INTEL® Z490 CHIPSET BLOCK DIAGRAM



Intelin Comet Lake (kuva IT World Canada)



Apple M1-yleiskuva (kuva Apple)



PROSESSIT

- Käyttöjärjestelmän ensimmäinen ja tärkein tehtävä oli siis koordinoida kaikkea käskyjen suoritusta, mitä tietokoneessa tapahtuu.
- Tämä on oikeassa elämässä vaikeaa
 - Meillä on useita käyttäjiä, useita ohjelmia, useita laitteita jne.
- Millä tällaista kaaosta hallitaan
 - Vastaus: Pilkkomalla monimutkaiset ongelmat pienempiin osiin
 - Kaikkien ongelmien kanssa ei tarvitse taistella samanaikaisesti
 - Kun meillä on n kappaletta pienempiä ongelmia, niin ratkotaan niitä yksi kerrallaan.



PROSESSIT

- Prosessi = Käyttöjärjestelmän tarjoama abstraktio, joka esittää kaikkea, mitä yksittäisen ohjelman ajamiseen tarvitaan
- Formaalisti: Prosessi on peräkkäinen jono käskyjen suoritusta omassa osoiteavaruudessaan.



PROSESSIT

- **Prosessissa on kaksi osaa:**
 - **Peräkkäinen suoritus**
 - Prosessissa ei ole mitään rinnakkaisuutta
 - kaikki suoritus on peräkkäistä, yksi käsky toisensa jälkeen
 - **Prosessin tila**
 - Kaikki muisti (muistipaikat), joihin prosessin suoritus vaikuttaa
 - rekisterit, keskusmuisti, tiedostot...
- **Prosessi ei ole sama asia kuin ohjelma!**



PROSESSIT

```
main(){
```

```
...
```

```
}
```

```
A() {
```

```
...
```

```
}
```

PROGRAM

```
main(){
```

```
...
```

```
}
```

```
A() {
```

```
...
```

```
}
```

heap

stack

A

main

registers, PC

PROCESS

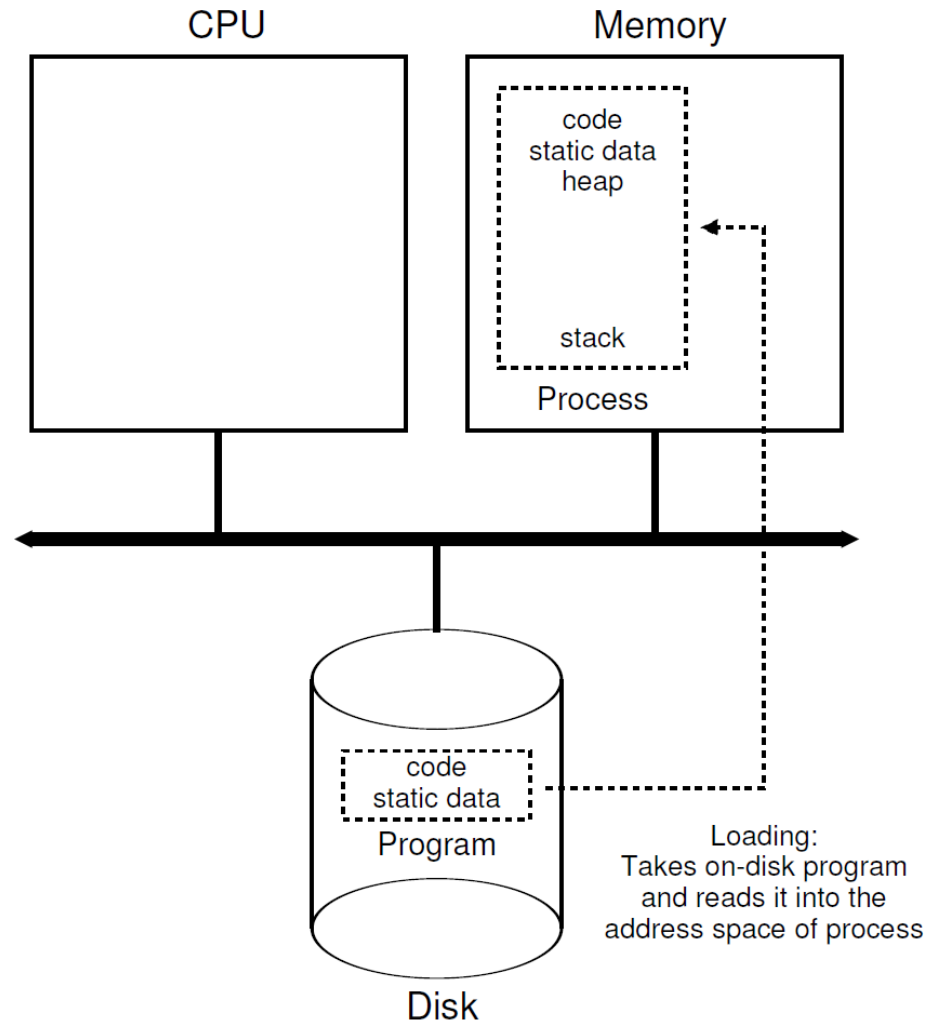


PROSESSIT

- Ohjelma on vain sarja komentoja ohjelmakoodissa
 - Mikä tahansa ohjelma, ls tai internet explorer
- Ohjelma on kyllä osa prosessia
 - Ohjelma on osa prosessin tilaa
 - Esim. 2 käyttäjää järjestelmässä
 - Minä ajan ohjelman, sinä ajat ohjelman. Sama ohjelma, eri prosessit!
- Ohjelma voi myös hierarkiassa olla prosessin yläpuolella
 - Ohjelma voi aloittaa uusia prosesseja eli käskeä järjestelmää suorittamaan muita ohjelmia



OHJELMA VS. PROSESSI



MÄÄRITELMIÄ

- **Yksiajo:** yksi prosessi on suorituksessa kerrallaan
 - Esimerkiksi sellaiset vanhat järjestelmät kuin MS/DOS tai Macintosh...
- käyttöjärjestelmän rakentamisen kannalta tämä olisi kaikkein kivointa
 - ei tarvitse taistella samanaikaisuuden kanssa ollenkaan
- PC:den kanssa idea oli, että yksi käyttäjä tekee yhtä asiaa kerrallaan
- Ei kuitenkaan kovin käyttäjäystävällistä?
 - Prosessien loppumista joutui(si) odottamaan
 - Esimerkiksi mitä järkeä olisi odottaa, että tulostustyö valmistuu, ennen kuin voi jatkaa tekstinkäsittelyä?



MÄÄRITELMIÄ

- **Moniajo:** Useita prosesseja voidaan ajaa samanaikaisesti
 - unix, OS/2, Windows 95...
- **Säie:** peräkkäinen jono käskyjä prosessin sisällä
 - mahdollistaa suorituksen samanaikaisuuden
 - Kevyempiä 'prosesseja' prosessin sisällä
 - Säikeillä ei ole omaa pääsyä käyttöjärjestelmän resursseihin, vaan ne jakavat niiden isäntäprosessien resurssit.



MÄÄRITELMIÄ

- **Muistiavaruus:** Kaikki tila, mitä ohjelman suoritus vaatii
 - Kirjaimellisesti kaikki keskusmuistin osat, mihin ohjelma pääsee kirjoittamaan.
 - Jokaisella ohjelmalla on oma muistiavaruutensa
 - ⇒ ‘näyttää’ siltä, kuin jokaista ohjelmaa ajettaisiin eri tietokoneessa (ohjelman suorituksen eristäminen, hiekkalaatikko, turvallisuus!)



MÄÄRITELMIÄ

- Kun puhutaan prosesseista käyttöjärjestelmän tasolla...
 - Kun puhutaan samanaikaisuudesta ja moniajosta, puhutaan oikeasti prosessien säikeistä ja niiden synkronoinnista
 - Kun puhutaan ohjelmien suorituksen turvallisuudesta tarkoitetaan nimenomaan prosessien muistiavaruuksia.



KÄYTTÖJÄRJESTELMÄN PÄÄOMINAISUUKSISTA

CT30A3370 - Käyttöjärjestelmät ja
systemiohjelmointi



KÄYTTÖJÄRJESTELMÄN 3 PÄÄOMINAISUUTTA

- Käyttöjärjestelmä on siis monimutkainen, mutta samalla keskeinen osa koko tietokonetta.
- Käytännössä käyttöjärjestelmälle voidaan asettaa kolme pääominaisuutta:
 - Virtualisointi (Virtualization)
 - Samanaikaisuus (Concurrency)
 - Pysyvyys (Persistence)
- Näitä ominaisuuksia käsittelee myös kurssikirja pääteemoinaan.



VIRTUALISOINTI

- Lyhyesti: ohjelmat saavat oman virtuaalisen prosessorin ja muistin.
- Ohjelmat toimivat omassa kehässään kiltisti tietämättä tai välittämättä siitä, että eivät ole ainoa käynnissä oleva prosessi.
- Käyttöjärjestelmän hommaksi jää huolehtia, että fyysiset resurssit jaetaan siten että virtualisointi toimii oikein.
- ...Ja tarjota rajapinta että ohjelmat ja laitteet pääsevät resursseihin käsiksi.



SAMANAIKAISUUS

- Lyhyesti: ohjelmat saadaan toimimaan samanaikaisesti jaetuilla resursseilla.
- Monisäikeiset (Multi-thread) ohjelmat ylipäänsä saadaan toimimaan, ja että rinnakaisuus on toteutettu järkevästi.
- Samanaikaisuus (concurrency) = Monta asiaa yhtä aikaa yhdessä suoritinresurssissa.
- Rinnakkaisuus (parallelism) = Monta samanaikaista prosessia joita ajetaan eri suoritinresursseilla.



PYSYVYYS

- Käyttöjärjestelmä mahdollistaa pysyvän tiedon tallettamisen ja palauttamisen.
- Käyttöjärjestelmän tulee siis pystyä käyttämään jotain levynhallintajärjestelmää (file system).
 - Ja tarjota palvelu jonka avulla tietoja voidaan liikutella edestakaisin.



MITÄ OMINAISUUKSIA KÄYTTÖJÄRJESTELMÄN SUUNNITTELUSSA VOISI OLLA?

- **Abstraktio:** laitteen käyttämiseen ja sille kehittämiseen tarjotaan helpottavia ja käteviä työkaluja.
 - Myös raudan mukaan; vrt. MISC, RISC ja CISC (minimal, reduced ja complex instruction set computer)
 - CISC nykyään tavallisin kotitietokoneessa, RISC suurteholaskennassa
- **Tehokkuus:** käyttöjärjestelmä minimoi oman suoritustarpeensa ja ottaa raudasta mahdollisimman paljon irti.
- **Suojaus, eristys:** käyttöjärjestelmä osaa suojata käynnissä olevia prosesseja viallisten tai pahantahtoisten prosessien sekoilulta tai kaappausyrityksiltä.



MITÄ OMINAISUUKSIA KÄYTTÖJÄRJESTELMÄN SUUNNITTELUSSA VOISI OLLA?

- **Luotettavuus:** Käyttöjärjestelmä ei tee mitään yllättävää tai poikkeavaa.
- **Energiatehokkuus:** mahdollistaa työskentelyn kevyemmällä rautakapasiteetilla.
- **Siirrettävyys:** maailma menee ja osittain on jo taskutietokoneiden aikakaudessa.

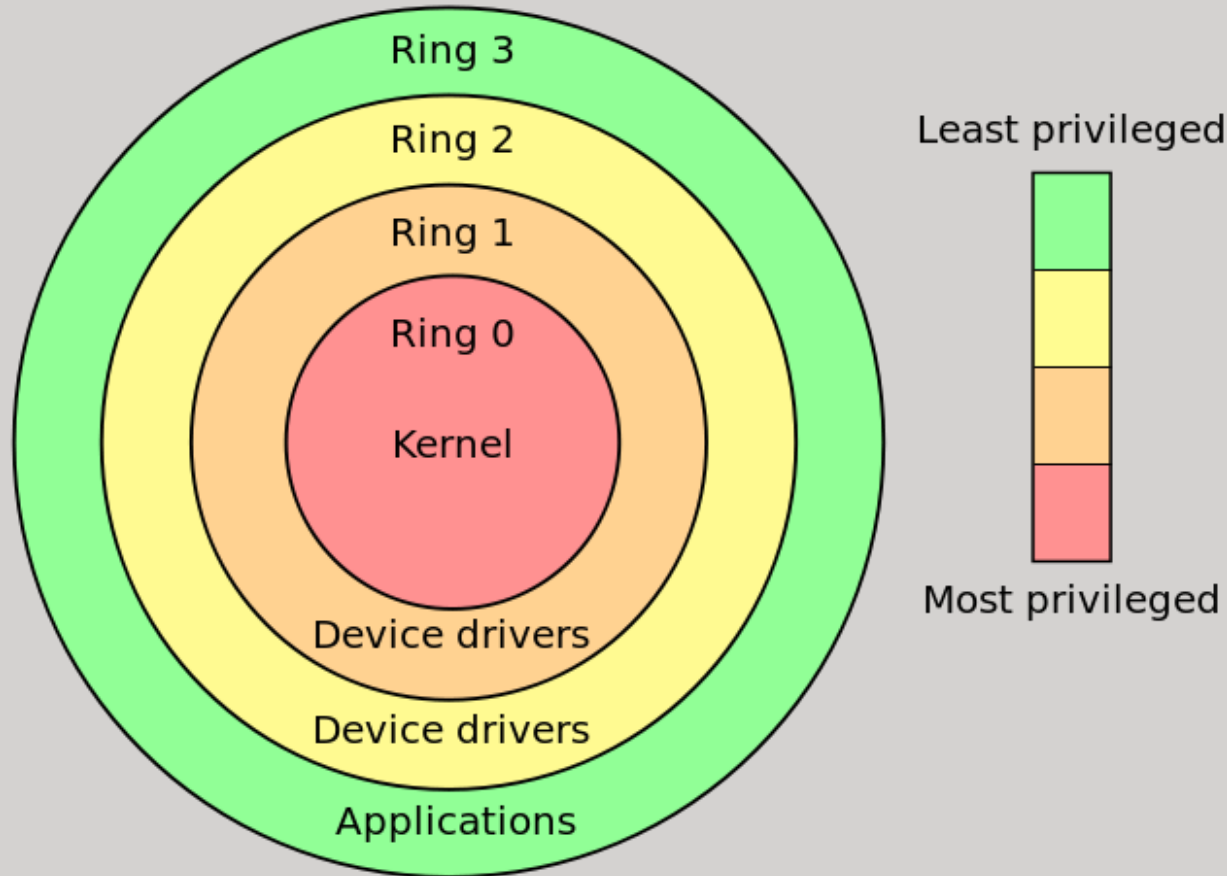


MITÄ OMINAISUUKSIA KÄYTTÖJÄRJESTELMÄN SUUNNITTELUSSA VOISI OLLA?

- **Vrt Minix 3, maailman tavallisin PC-käyttöjärjestelmä.**
 - **Kiitos Intelin "management systemin" joka elää kokonaan prosessorin pään sisällä.**



X86-PROSESSIEN PRIVILEEGIOTASOT



The ME is colloquially categorized as ring -3, below [System Management Mode](#) (ring -2) and the [hypervisor](#) (ring -1), all running at a higher privilege level than the kernel (ring 0)

Kuva: [CC BY-SA 3.0](#)
Wikipedia



MITÄ OMINAISUUKSIA KÄYTTÖJÄRJESTELMÄN SUUNNITTELUSSA VOISI OLLA?

- **Vrt. ARM –prosessoriperhe**

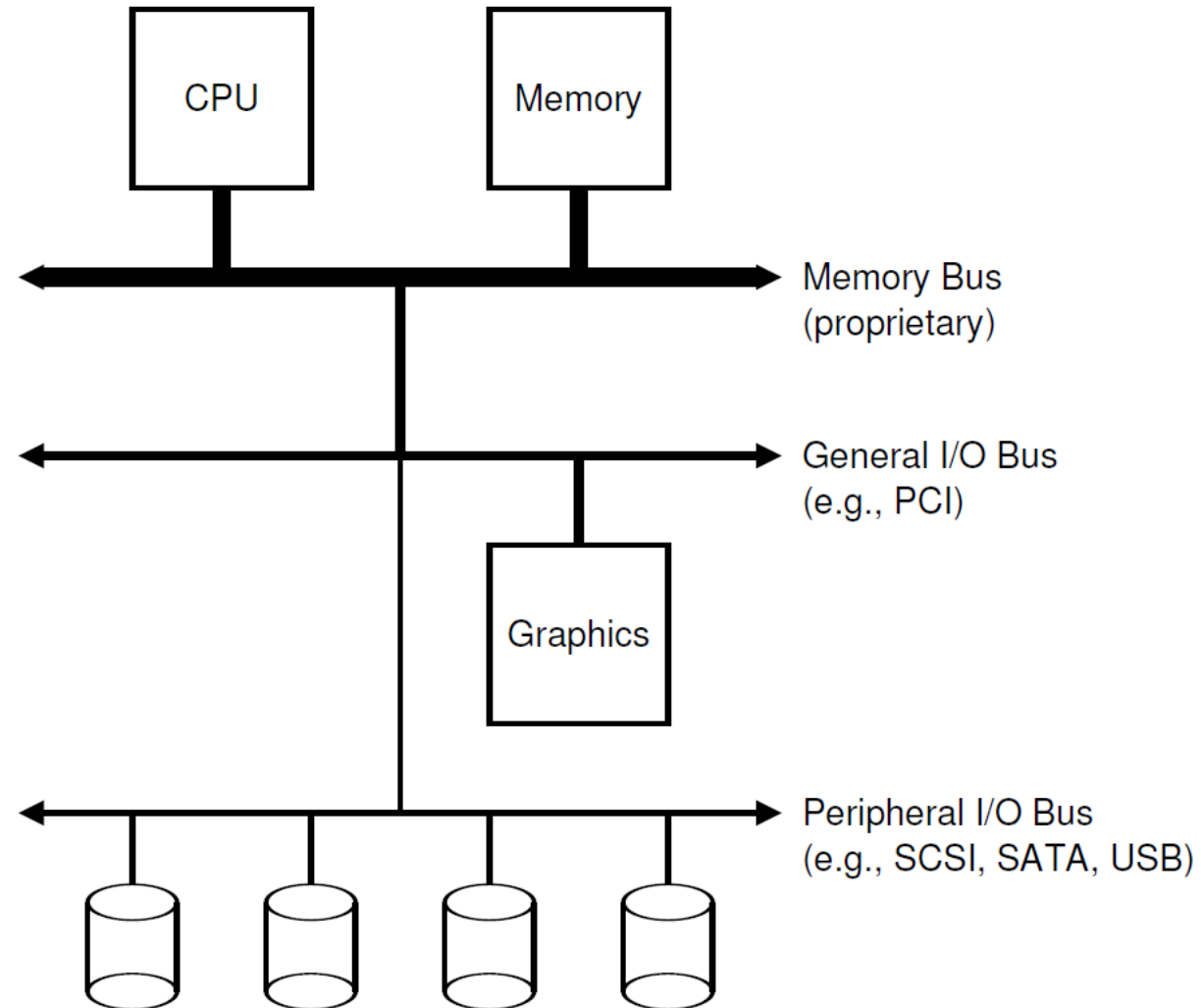
- Valmistettu vuodesta 1985
- Uusin malli ARMv8.4

- **Valmistettu arviolta 100 miljardia kappaletta.**

” Intel makes roughly 300-400 million CPUs each year. This might be a big number for one company but its is less than 1% of the total microprocessor market.”



VRT: NYKYINEN TIETOKONE PERIAATTEELLISELLA TASOLLA



ESIMERKKI: XV6

- xv6 on moderni versio Unixin 6. versiosta, suunniteltu toimimaan x86-prosessoreilla.
- Käytetään joissain harjoitustehtävissä ja luentoexamplesimerkkeinä sopivissa tilanteissa.
- Ei osaa mitään jännää, mutta boottaa ja sisältää jotain perustoimintoja.
 - Koko käyttiksen koko alle 10k riviä koodia.
 - Koko lähdekoodi tulostettavana manuaalina
 - Normaalit modernit käyttöikset miljoonia rivejä.

<https://github.com/mit-pdos/xv6-public>



MITÄ OMINAISUUKSIA KÄYTTÖJÄRJESTELMÄN SUUNNITTELUSSA VOISI OLLA?

- Käyttöjärjestelmä ei välttämättä ole monimutkainen ohjelma.
 - Ei ole yhtä be all – catch all –käyttöjärjestelmää.
 - Linux/Windows 50-60 miljoonaa koodiriviä, tavoitteena monikäyttöisyys ja mukautuvuus eri alustoihin.
- Jos näitä vaatimuksia ei olisi, voi käyttöjärjestelmä olla alle 10k riviä koodia.
 - Windows 3.1 2.5 miljoonaa riviä koodia n. 200 työntekijältä.
 - Ja sitä käytetään edelleen!



MITÄ TÄSTÄ LUENNOSTA PITÄÄ MUISTAA?

- Käyttöjärjestelmän määritelmä
- Prosessi vs. ohjelma
- Käyttöjärjestelmän ominaisuudet



HARJOITUKSET

- Ekat harjoitukset tällä viikolla; lukekaa tehtävät läpi, kokeilkaa niitä itse ja menkää harjoituksiin jos tarvitsette apua.
 - Osaan tehtävistä lisämateriaalia, apua kurssikirjasta.





LUT
University