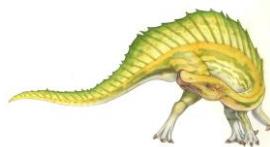
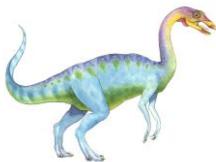


2. Operacinių sistemų struktūra 1





2 skyriaus temos

- Operacinių sistemų paslaugos
- Vartotojo-OS sąsaja
- Sisteminiai iškvietimai
- Sisteminių iškvietimų tipai
- Sisteminės programos
- OS projektavimas ir įgyvendinimas
- OS struktūra
- OS klaidų paieška

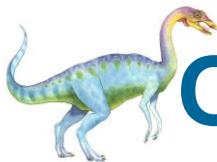




Tikslai

- Apibūdinti paslaugas, kurių OS suteikia vartotojams, procesams bei kitoms sistemoms.
- Aptarti skirtingus OS struktūrizavimo būdus.

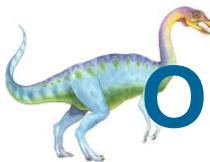




Operacinių sistemų paslaugos

- OS suteikia programų ir servisų paleidimo ir vykdymo aplinką programoms ir vartotojams.
- Vienas iš OS servisų rinkinių suteikia vartotojui reikalingas paslaugas:
 - **Vartotojo sąsają** – beveik visos OS turi vartotojo sąsają (**UI – user interface**).
 - ▶ Ji varijuoja: **Command-Line (CLI)**, **Graphics User Interface (GUI)**, **Batch**
 - **Programų vykdymą** – sistema turi gebeti paleisti programą į atmintį ir ją vykdyti.
 - **Įvesties/išvesties operacijas** – vykdomai programai gali reikėti tam tikrų I/O, įskaitant darbą su failais ar I/O įrenginiais.





Operacinių sistemų paslaugos (tęs.)

- Vienas iš OS servisų rinkinių suteikia vartotojui reikalingas paslaugas (tęs.):
 - **Failų-sistemos manipuliavimą** – programoms reikia skaityti iš failų ir rašyti į failus bei katalogus, kurti bei trinti juos, ieškoti, skaityti failo metaduomenis, valdyti prieigos leidimus.
 - **Komunikaciją** – procesai gali apsikeisti informacija tame pačiame kompiuteryje ar tarp skirtinų kompiuterių per tinklą.
 - ▶ Komunikacija gali būti vykdoma per bendrą (angl. *Shared*) atmintį arba perduodant žinutes.
 - **Klaidų aptikimą** – OS nuolat turi suvokti galimas klaidas, kurios gali įvykti:
 - ▶ Procesoriuje ir atminties techninėje įrangoje, I/O įrenginiuose bei vartotojo programose.
 - ▶ Kiekvienam klaidos tipui, OS turi priimti tinkamus veiksmus, kad užtikrinti teisingą ir nuoseklų darbą.
 - ▶ *Debugging* galimybės gali gerokai padidinti vartotojo ir programuotojo galimybes efektyviai naudotis sistema.

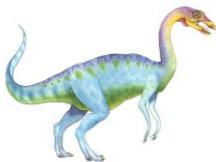




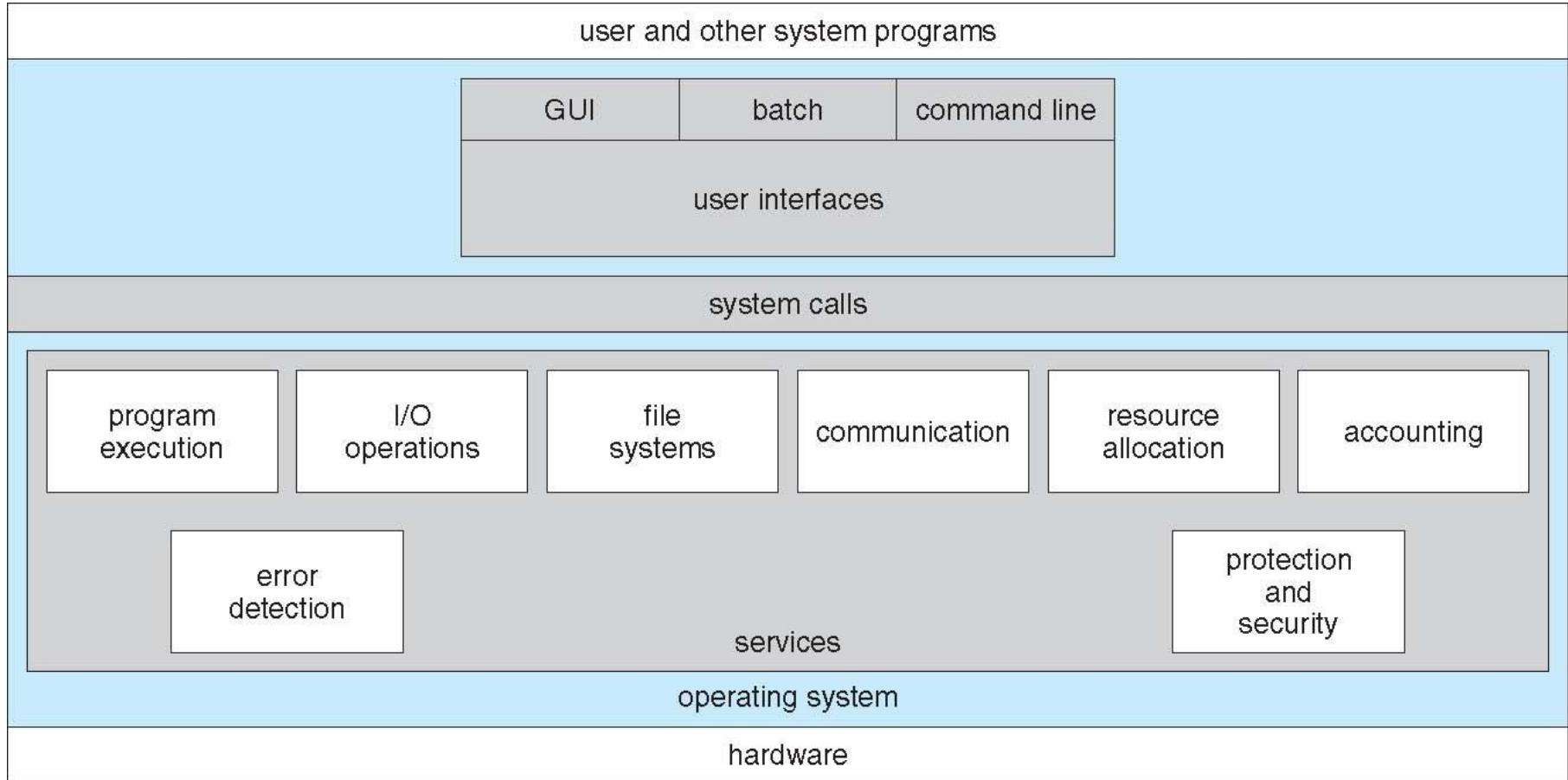
Operacinių sistemų paslaugos (tęs.)

- Kitas OS funkcijų rinkinys reikalingas užtikrinti efektyvų sistemos veikimą per resursų pasidalijimą:
 - **Resursų paskirstymui** – kai keli vartotojai ar kelios užduotys vykdomos lygiagrečiai, resursai turi būti priskiriami abiems - CPU ciklai, pagrindinė atmintis, failų saugyklos, I/O įrenginiai.
 - **Apskaitai** – kad žinoti kurie vartotojai (programos) naudoja kiek ir kokių kompiuterio resursų.
 - **Apsauga ir saugumas** – informacijos, saugomos keleto vartotojų ar tinklinėje sistemoje savininkai turi galėti kontroliuoti šią informaciją, lygiagretūs procesai neturi interferuoti vieni su kitais.
 - ▶ **Apsauga** – užtikrinimas, kad prieiga prie sistemos resursų yra kontroliuojama
 - ▶ **Saugumas** – vartotojai iš išorės turi autentikuotis, išorinių I/O įrenginių apsaugojimas nuo neteisingos prieigos bandymų.





OS paslaugų bendras vaizdas





Vartotojo-OS sasaja - CLI

CLI ar **command interpreter** leidžia tiesioginį komandų įvedimą

- Kartais įgyvendinama branduolyje, kartais sistemonėje programoje.
- Kartais įgyvendinama, kaip **shells**.
- Pagrinde, paima komandas iš vartotojo ir jas paleidžia.
- Kartais, komandos yra integruotos, kartais tai tik programų vardai
 - ▶ Pastaruoju atveju, naujų galimybių pridėjimui nereikia modifikuoti shell'o





Bourne Shell Command Interpreter

Default

New Info Close Execute Bookmarks

Default Default

```
PBG-Mac-Pro:~ pbgs w
15:24 up 56 mins, 2 users, load averages: 1.51 1.53 1.65
USER    TTY      FROM          LOGIN@  IDLE WHAT
pbgs   console -          14:34      50 -
pbgs   s000   -          15:05      - w

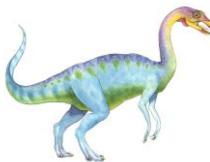
PBG-Mac-Pro:~ pbgs iostat 5
          disk0           disk1           disk10          cpu      load average
          KB/t  tps  MB/s   KB/t  tps  MB/s   KB/t  tps  MB/s  us sy id  1m   5m   15m
  33.75 343 11.30   64.31 14  0.88   39.67  0  0.02 11  5 84  1.51 1.53 1.65
  5.27 320 1.65   0.00  0  0.00   0.00  0  0.00  4  2 94  1.39 1.51 1.65
  4.28 329 1.37   0.00  0  0.00   0.00  0  0.00  5  3 92  1.44 1.51 1.65

^C
PBG-Mac-Pro:~ pbgs ls
Applications           Music           WebEx
Applications (Parallels) Pando Packages config.log
Desktop                 Pictures         getsmartdata.txt
Documents               Public          imp
Downloads              Sites           log
Dropbox                 Thumbs.db       panda-dist
Library                Virtual Machines prob.txt
Movies                 Volumes         scripts

PBG-Mac-Pro:~ pbgs pwd
/Users/pbg

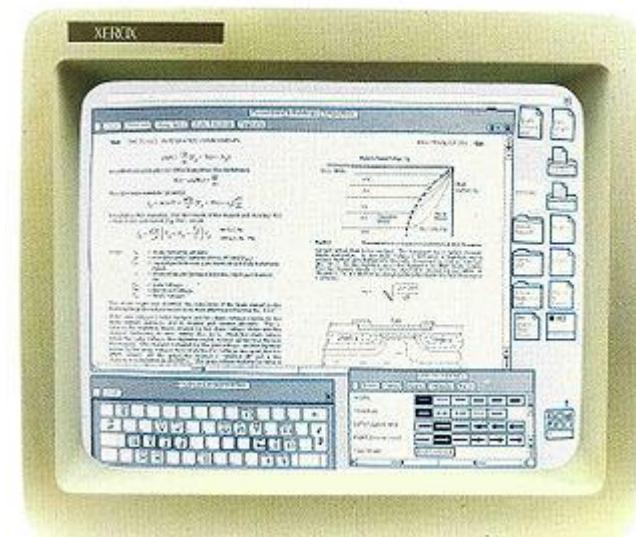
PBG-Mac-Pro:~ pbgs ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=2.257 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.262 ms
^C
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.262/1.760/2.257/0.498 ms
PBG-Mac-Pro:~ pbgs 
```





Vartotojo-OS sasaja - GUI

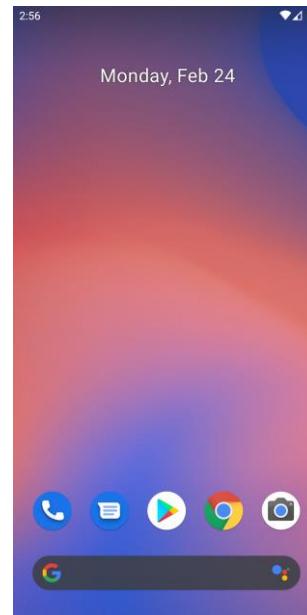
- Vartotojui draugiška **desktop** metaforų sasaja
 - Paprastai – pelė, klaviatūra, monitorius
 - **Ikonos** atvaizduoja failus, programas, veiksmus ir kt.
 - Skirtingi pelės mygtukai virš objektų sasajoje sukelia skirtingus veiksmus (suteikti informaciją, opcijas, paleisti funkcijas, atidaryti katalogus)
 - Išrastas Xerox PARC centre 1981 m. Xerox 8010 Star Information System
- Dauguma šių laikų sistemų dabar turi ir CLI ir GUI sasajas:
 - Microsoft Windows yra GUI su CLI “command” shell
 - Apple Mac OS X „Aqua” GUI sasaja su UNIX (branduoliu) kernel ir prieinamu shell.
 - Unix ir Linux turi CLI su pasirinktine GUI sasaja (CDE, KDE, GNOME, XFCE, LXDE ir kt.)



Liečiamujų ekrano sasajos

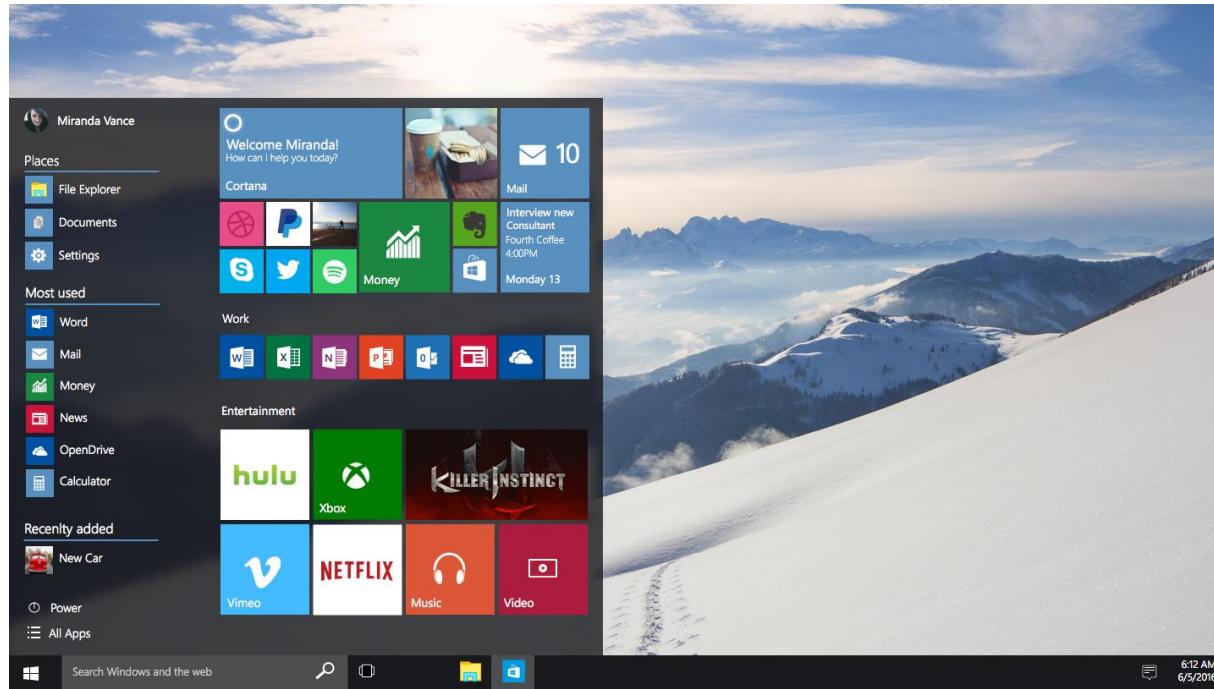
nJrenginiams su liečiamaisiais ekranais reikia naujų sasaju

- | Néra pelēs arba ji nenumatyta
- | Veiksmai ir pasirinkimai grindžiami gestais
- | Teksto įvesčiai naudojama virtuali klaviatūra
- | Balso komandos.

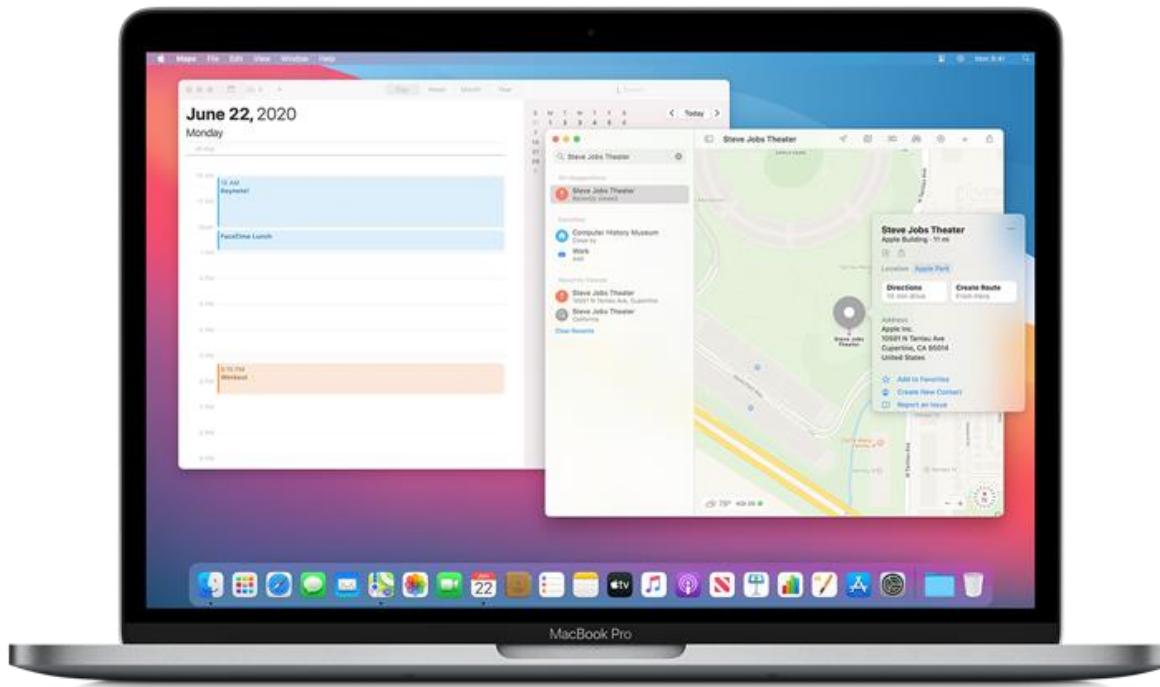


<https://www.apple.com/lt/newsroom/2021/06/ios-15-brings-powerful-new-features-to-stay-connected-focus-explore-and-more/>

Windows UI

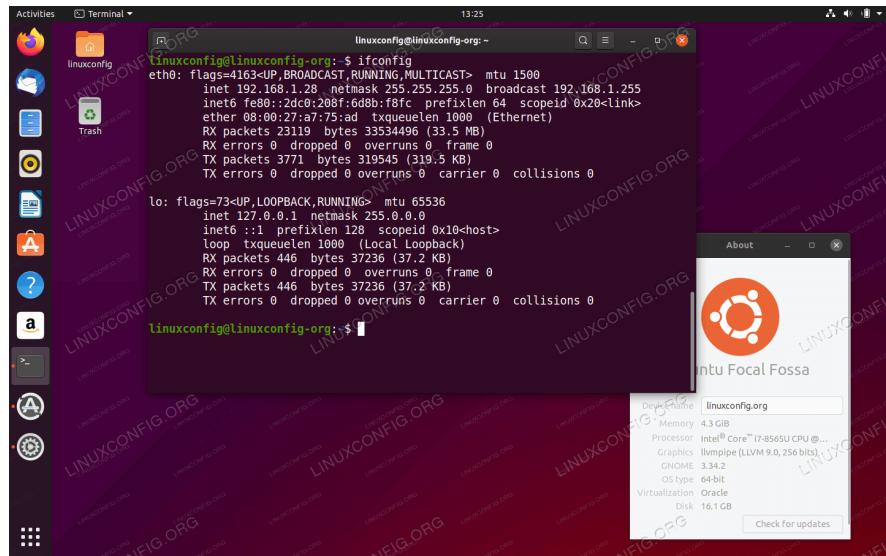


MacOS UI



<https://developer.apple.com/design/human-interface-guidelines/macos/overview/whats-new-in-macos/>

Ubuntu/Deepin UI

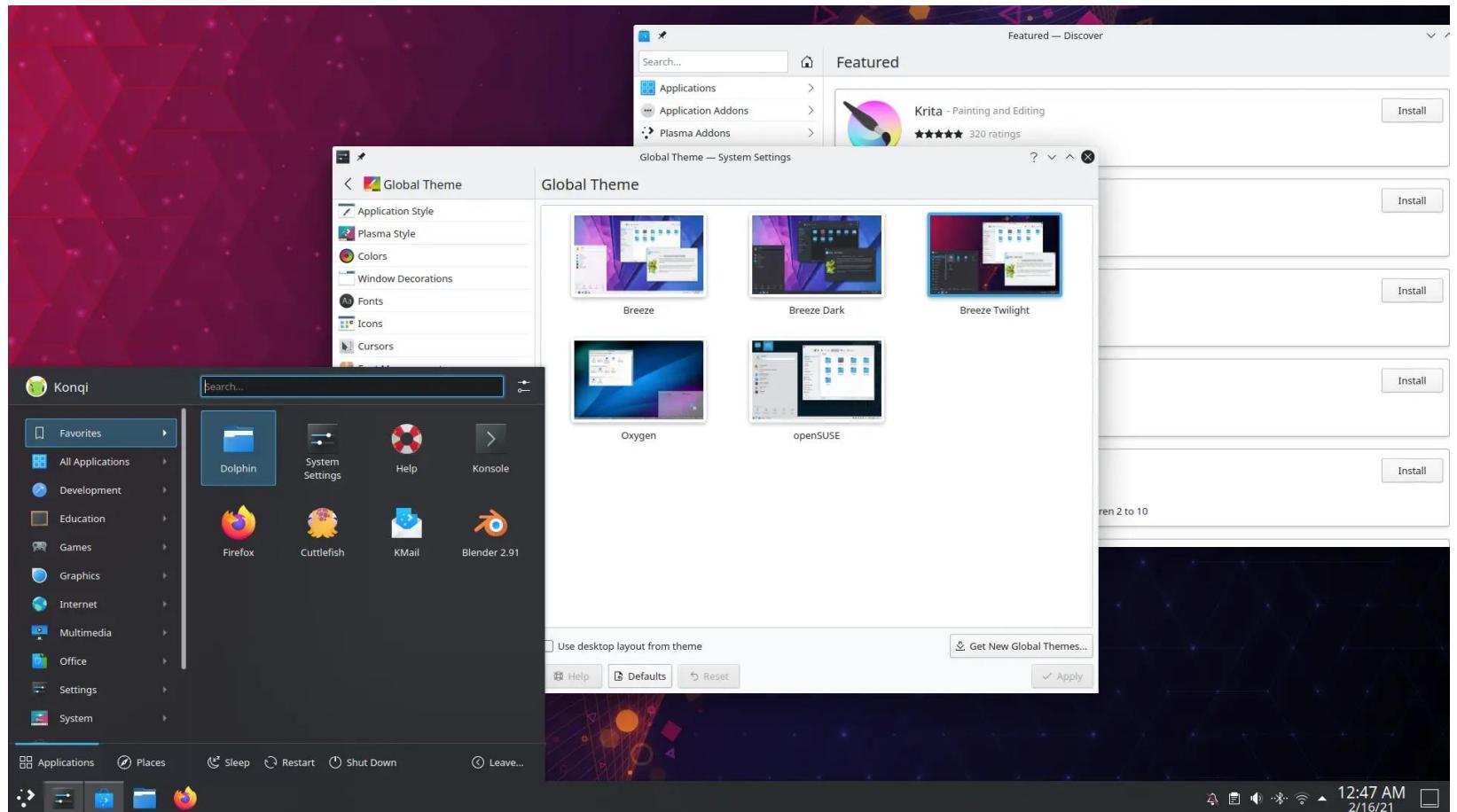


<https://linuxconfig.org/how-to-switch-back-networking-to-etc-network-interfaces-on-ubuntu-20-04-focal-fossa-linux>

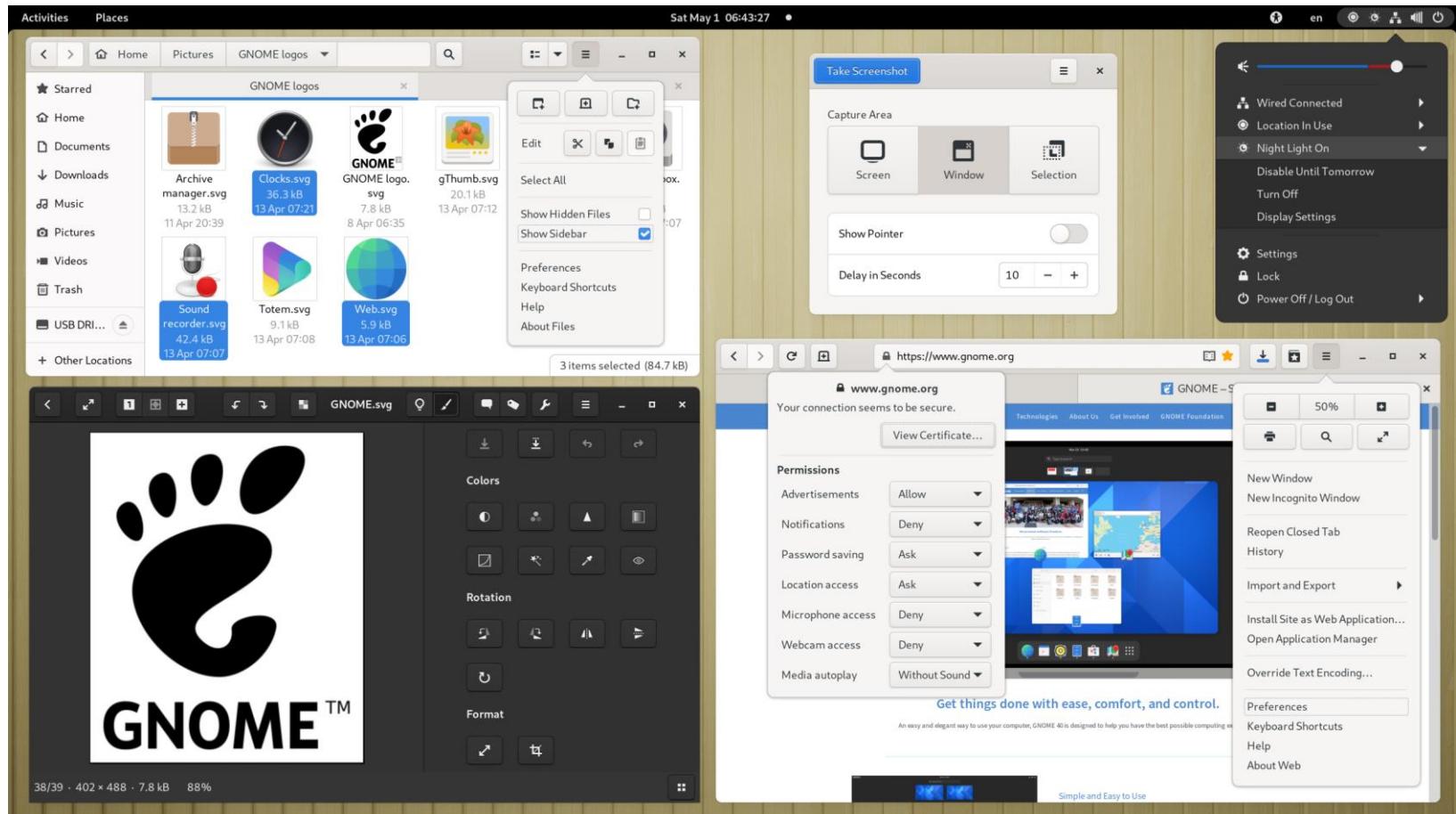


https://news.softpedia.com/news/gorgeous-deepin-15-2-linux-os-adopts-new-launcher-interface-intuitive-search-504721.shtml#sgal_0

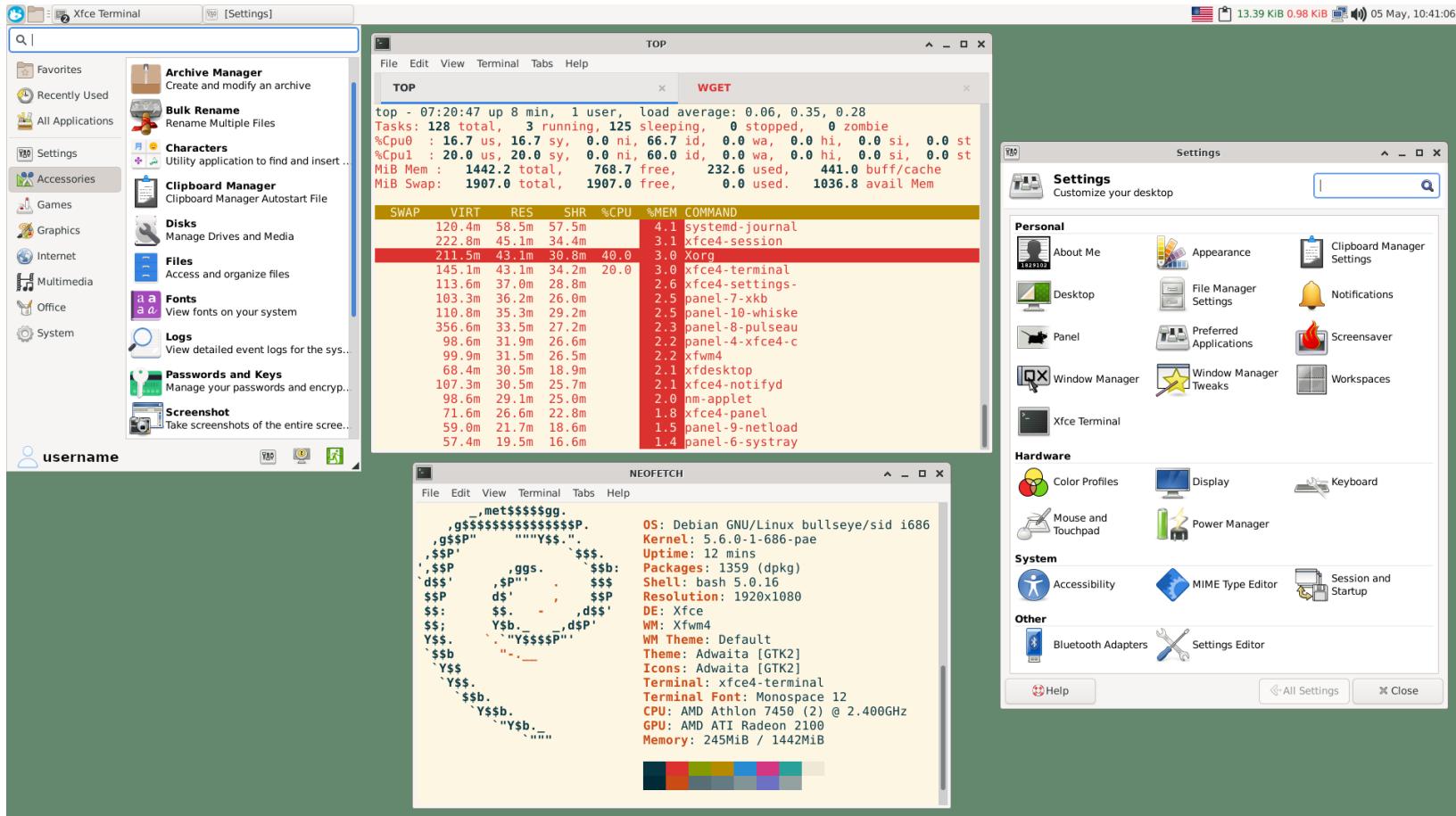
KDE Plasma



GNOME



Xfce



iPadOS UI



<https://edu.gcfglobal.org/en/ipadbasics/using-ios/1/>



<https://www.apple.com/newsroom/2019/06/the-new-ipados-powers-unique-experiences-designed-for-ipad/>

Command line interface

```
PS C:\> Get-ChildItem 'MediaCenter:\Music' -rec |  
    >>     where { -not $_.PSIsContainer -and $_.Extension -match 'wma|mp3' } |  
    >>     Measure-Object -property length -sum -min -max -ave  
  
Count : 1307  
Average : 549126.0953887  
Sum : 74097857  
Maximum : 22905267  
Minimum : 3236  
Property : Length  
  
PS C:\> Get-WmiObject CIM_BIOSElement | select biosu*, man*, ser* | Format-List  
  
BIOSVersion : <IOSCPL - 6040000, Ver 1.00PARTIBL>  
Manufacturer : TOSHIBA  
SerialNumber : M82ii16H  
  
PS C:\> <#imiSearcher|>  
    >> SELECT * FROM CIM_Job  
    >> WHERE Priority > 1  
    >> '$e'.get() | Format-Custom  
  
class ManagementObject#root\cimv2\Win32_PrintJob  
<  
  Document = Monad Manifesto - Public  
  jobID = 6  
  JobStatus =  
  Owner = User  
  Priority = 42  
  Size = 1027088  
  Name = Epson Stylus COLOR 740 ESC/P 2, 6  
>  
  
PS C:\> $rssUrl = 'http://blogs.msdn.com/powershell/rss.aspx'  
PS C:\> $blog = [xml]::new().CreateObject('System.Net.WebClient').DownloadString($rssUrl)  
PS C:\> $blog.rss.channel.item | select title -first 3  
title  
---  
MMS: What's Coming In PowerShell 02  
PowerShell Presence at MMS  
MMS Talk: System Center Foundation Technologies  
  
PS C:\> $host.version.ToString().Insert(0, 'Windows PowerShell: ')  
Windows PowerShell: 1.0.0.0  
PS C:\>
```

Terminal

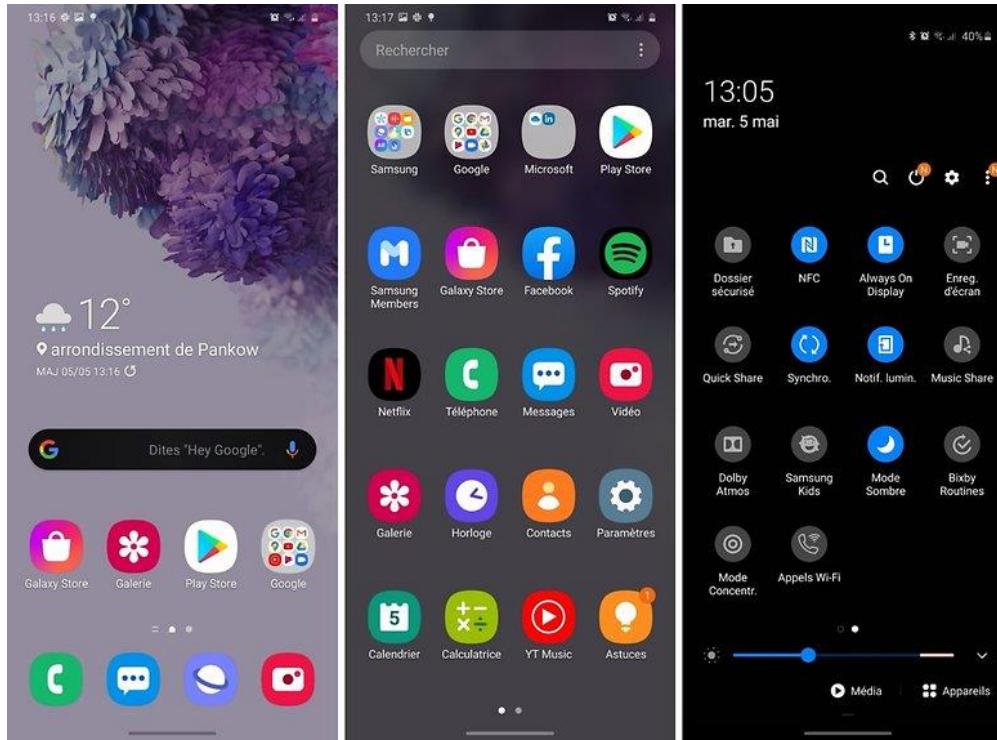
File	Mode	Owner	Group	Last Modified	Size	Type
-rwxr-xr-x 1 bin	18296	Jun 8	1979	fsck		
-rwxr-xr-x 1 bin	1458	Jun 8	1979	getty		
-rw-r--r-- 1 root	49	Jun 8	1979	group		
-rwxr-xr-x 1 bin	2482	Jun 8	1979	init		
-rwxr-xr-x 1 bin	8484	Jun 8	1979	mkfs		
-rwxr-xr-x 1 bin	3642	Jun 8	1979	mknod		
-rwxr-xr-x 1 bin	3976	Jun 8	1979	mount		
-rw-r--r-- 1 root	141	Jun 8	1979	passwd		
-rw-r--r-- 1 bin	366	Jun 8	1979	rc		
-rw-r--r-- 1 bin	266	Jun 8	1979	ttys		
-rwxr-xr-x 1 bin	3794	Jun 8	1979	umount		
-rwxr-xr-x 1 bin	634	Jun 8	1979	update		
-rw-r--r-- 1 bin	40	Sep 22	05:49	utmp		
-rwxr-xr-x 1 root	4520	Jun 8	1979	wall		
# ls -l /sunix*						
-rwxr-xr-x 1 sys	53302	Jun 8	1979	/hphtunix		
-rwxr-xr-x 1 sys	52850	Jun 8	1979	/hptmunix		
-rwxr-xr-x 1 root	50990	Jun 8	1979	/rkunix		
-rwxr-xr-x 1 root	51982	Jun 8	1979	/rl2unix		
-rwxr-xr-x 1 sys	51790	Jun 8	1979	/rphtunix		
-rwxr-xr-x 1 sys	51274	Jun 8	1979	/rptmunix		
# ls -l /bin/sh						
-rwxr-xr-x 1 bin	17310	Jun 8	1979	/bin/sh		
#						

iPhone UI



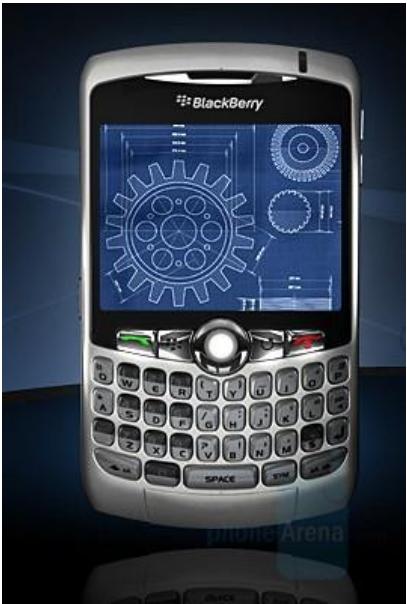
<https://9to5mac.com/2020/10/15/how-to-use-compact-iphone-call-interface-ios-14/>

Samsung S21 UI



<https://licensetoblog.com/samsung-galaxy-s20-plus-review-a-flagship-that-ticks-all-the-right-boxes/>

Geriausi išmanieji telefonai iki 2007



Say hello to iPhone.



https://www.phonearena.com/news/These-were-the-best-phones...before-the-iPhone_id95512

before 2007



after 2007



Samsung Touchscreen Tablet BEFORE iPad	Apple's iPad 2 (announced March 2011)	Samsung Touchscreen Tablet AFTER iPad 2
		

Phones before iPhone X



Phones after iPhone X



https://www.reddit.com/r/iphone/comments/abd5he/its_not_just_a_fruit_phone/



<https://www.wearable.com/apple/best-apple-watch-apps-832>



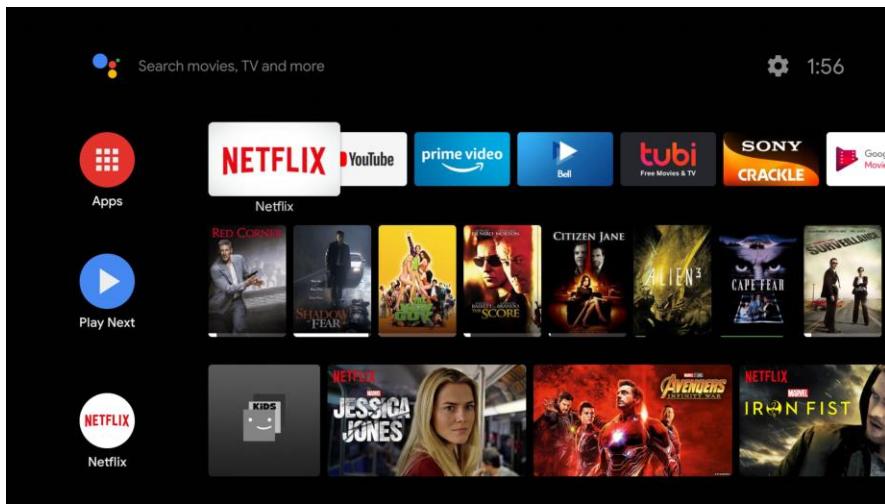
https://www.gsmarena.com/samsung_galaxy_watch_active_running_one_ui_shown_in_new_renders-news-35491.php



<https://www.giztop.com/xiaomi-mi-band-5-global.html>



<https://apps.garmin.com/pt-BR/apps/a58303e8-b0a8-4387-ba1c-519a648d3b70>



<https://9to5google.com/2020/05/06/android-tv-interface-opinion/>



<https://news.samsung.com/us/samsung-electronics-introduces-advanced-smart-tv-user-experience/>



<https://ifworlddesignguide.com/entry/107494-lg-smart-tv-gui>



<https://martechtoday.com/say-hello-to-voice-interface-optimization-216626>

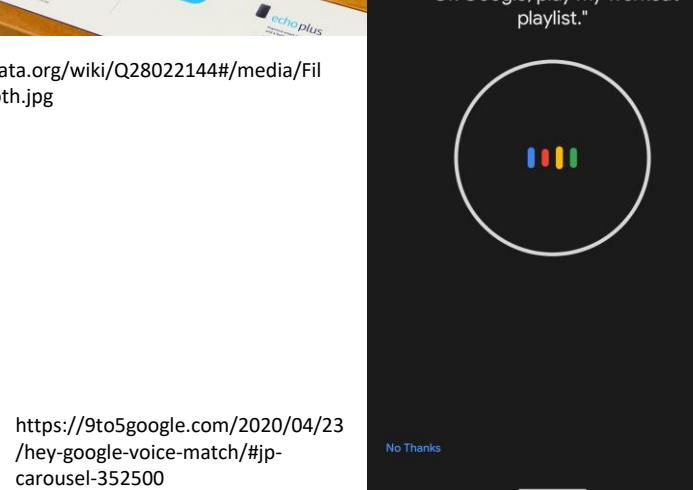


<https://www.wikidata.org/wiki/Q28022144#/media/File:AmazonAlexaBooth.jpg>



Hey Siri

<https://www.macrumors.com/2019/07/26/siri-human-analysis-voice-recordings/>



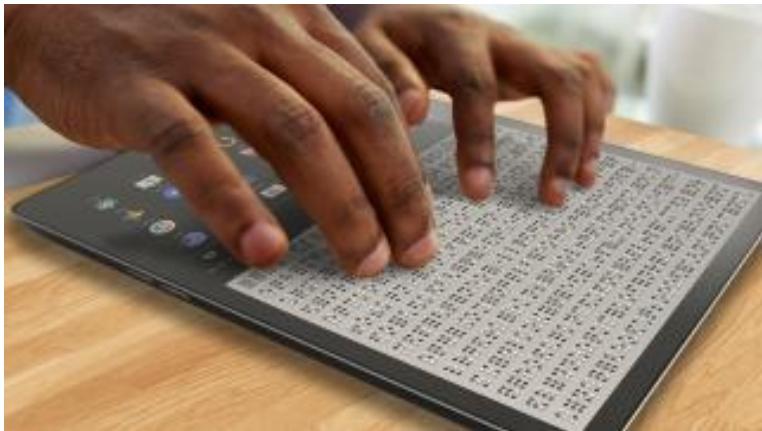
<https://9to5google.com/2020/04/23/hey-google-voice-match/#jp-carousel-352500>



<https://www.amikubota.com/ui-challenge>



<https://www.gbnews.ch/accessibility-for-all/>



<https://www.oho.com/blog/ada-website-compliance-preparing-your-college-or-university-website>



<https://adcycle.co/insights/>



Computer – brain interface



<https://wiharper.com/brain-computer-interface-company-neuracle-raises-rmb-60-million/>



WEARABLE COMPUTING

life style is changing...

3366 - DIAGONAL-TREBALL 242m
4398 - RAMON TURRO ... 265m
630 - BAC DE RODA... 268m
1161 - PARC CENTRAL P... 456m
5868 - DIAGONAL MAR-L... 525m
3261 - LLULL-LOPE DE ... 560m

547 m OF: 3261
717 m OF: 3168
433 m OF: 0108

3366 - DIAGONAL-TREBALL 242m
Dirección: AV. DIAGONAL, 72
Teléfono: 932299730
Cajeros: 2

1/2



Jayran Narayanan. Injecting a brand personality! UX & UI

2D SENSING – SURFACES

Human computer interaction: surface and gesture interfaces



Varun Narayanan, Injecting a brand personality! UX & UI

3D SENSING – GESTURES

3D Depth sensing, Face and body tracking solutions



Jayran Narayanan. Injecting a brand personality! UX & UI



Space shuttle Atlantis UI, (© collectSPACE/Robert Z. Pearlman)



The control room of the nuclear ship NS Savannah





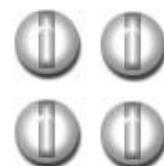








<https://ellis.fyi/blog/why-do-most-microwaves-have-such-a-terrible-user-interface/>





<https://shelterarchitecture.com/introducing-design-for-all/>

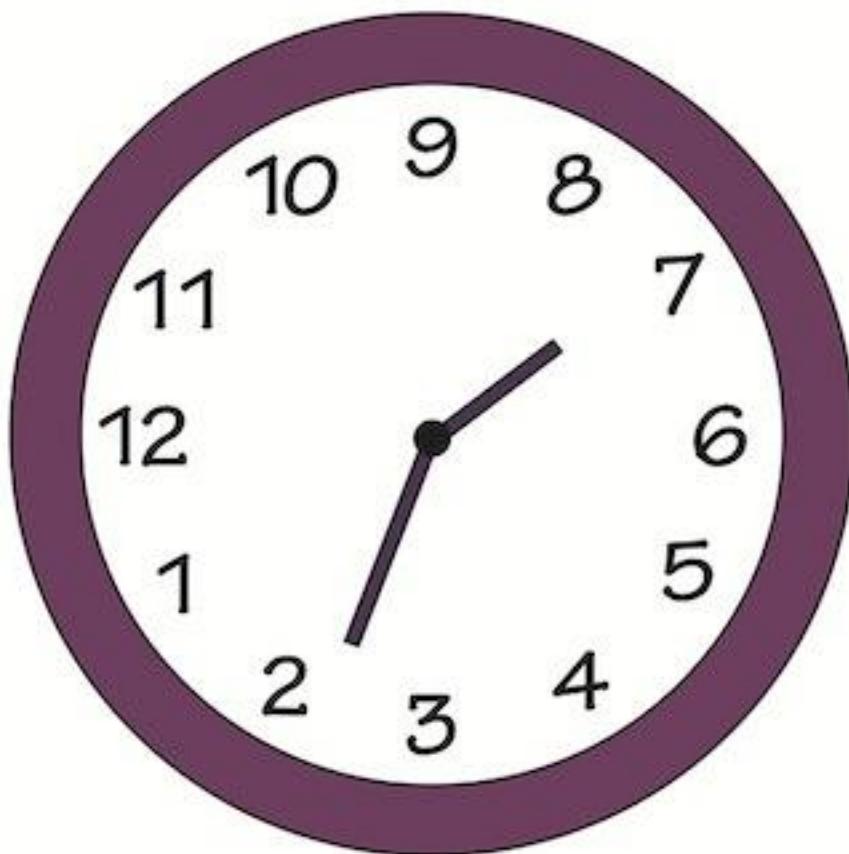
The thematic report / Teminė ataskaita

Universal design
Clarification of the concept

Universalus dizainas
Savokos išaiškinimas

MILJØVERNEDEPARTEMENTET
The Norwegian Ministry of the Environment

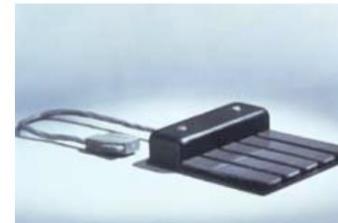
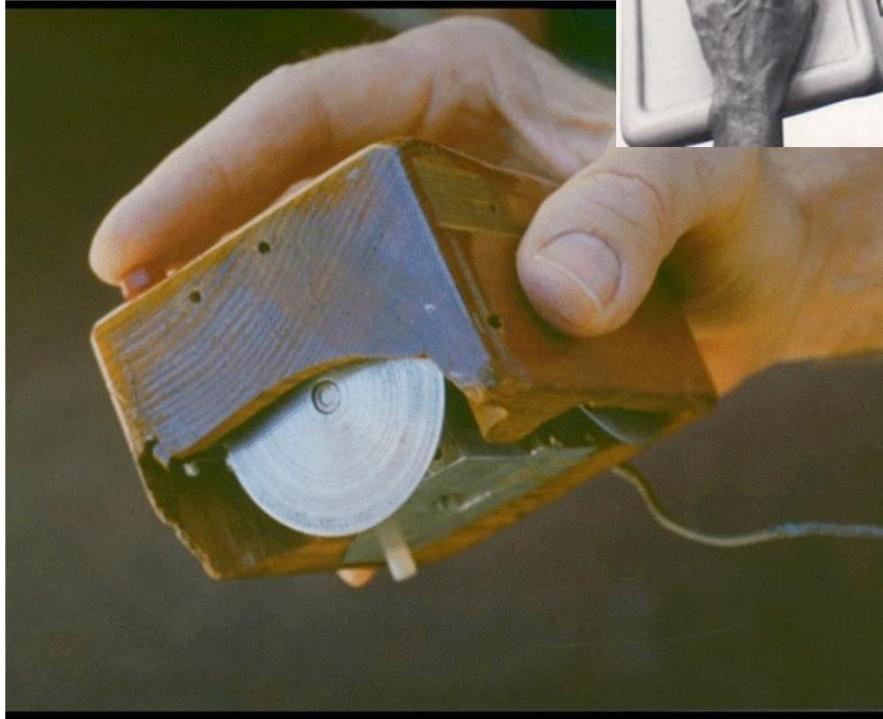
<https://www.yumpu.com/en/document/read/38839433/universal-design-universalus-dizainas-universell-utforming>



Nonstandard clock. “This clock is just as logical as the standard one, except the hands move in the opposite direction and ‘12’ is not in its usual place. Same logic, though. So why is it so difficult to read? What time is being displayed? 7:11, of course.” (From chapter 6 of *The Design of Everyday Things*: Revised and Expanded Edition, courtesy of Basic Books)

1968

Doug Engelbart



Clecio Bachini, Rafael Burity. From HCI to UX: Building a New meaning through the history in the industry



Xerox Alto
1973

System Browser

Collections-Bag	Interval	accessing	collect:
Collections-Text	LinkedList	copying	do:
Collections-Array	MappedCollection	adding	do:andBetweenDo:
Collections-Stream	OrderedCollection	removing	promoteFirstSuchT
Collections-Support	SortedCollection	enumerating	reverse
Graphics-Primitives		private	select:
Graphics-Display			Form Editor
Graphics-Media			
Graphics-Paths			

instance class

collect: aBlock
"Evaluate aBlock with each of my elements as the argument. Collect resulting values into a collection that is like me. Answer with collection. Override superclass in order to use add:, not at:put."

```
| newCollection |
newCollection := self species new.
self do: [:each | newCollection add: (aBlock value: each)].
^newCollection
```

User Interrupt

```
Paragraph>char: aCharBlockAtPoint:
Paragraph>processSelects:
CodeController(ParagraphEditor)>processRedButton
CodeController(ParagraphEditor)>processMouseButtons
CodeController(ParagraphEditor)>controlActivity
CodeController(Controller)>controlLoop
```

controlActivity

```
self scrollBarContainsCursor
ifTrue:
  [self scroll]
ifFalse:
  [self processKeyboard
  self processMouse]
```

blueButton 31@507 corner: 63@770
scrollBar marker savedArea paragraph startBlock

ScreenForm

- [None]<(Robson)SF>*
- [None]<(Robson)SF>ScreenForm.st
- [None]<(Robson)SF>ScreenFormChanges.st
- [None]<(Robson)SF>WordGraphics.form

Rectangle fromUser origin

ScreenForm setFullPageWidth.

(Form readFrom: 'FilledSkate.form') edit

Fig.1

Clecio Bachini, Rafael Burity. From HCI to UX: Building a New meaning through the history in the industry

1983

Ben Shneiderman - Direct Manipulation

- 1983 coined the term
Direct Manipulation

Direct manipulation systems offer the satisfying experience of operating on visible objects. The computer becomes transparent, and users can concentrate on their tasks.

SPECIAL FEATURE

Direct Manipulation: A Step Beyond Programming Languages

Ben Shneiderman, University of Maryland

Lithioir sought to make the form of a symbol reflect its content. "In signs," he wrote, "one sees an advantage for directness, for immediacy, for transparency. The true nature of a thing briefly said, as it were, pictures it; then, indeed, the later thought is wonderfully diminished."

Frederick K. Mather, "The Art of Symbols," Scientific American, May 1968

Certain interactive systems generate glowing enthusiasm among users—in marked contrast with the more common reaction of grogging acceptance or outright hostility. The enthusiastic user's reports are filled with positive feelings regarding

- mastery of the system
- ease in learning the performance of their task
- ease in learning the system originally and in assimilating advanced features,
- confidence in their capacity to retain mastery over time
- enjoyment in using the system,
- eagerness to show it off to novices, and
- desire to explore more powerful aspects of the system.

These feelings are not, of course, universal, but the analogous ones convey an image of the truly pleasant. As I talked with users of various systems and learned the terms they used, I began to develop a model of the features that produced such delight. The central ideas seemed to be visibility of the object of interest; rapid, reversible, and direct actions; and replacement of complex command language syntax by direct manipulation of the object of interest—hence the term “direct manipulation.”

Display editors. “Once you’ve used a display editor, you’ll never want to go back to a line editor. You’ll be spoiled.” This reaction is typical of those who use full-page display editors, who are great advocates of their systems. It is also typical of the enthusiastic comments from users of stand-alone word processors such as the Wang system and from users of display editors such as EMACS on the MIT/Hewlett-Packard Multics system and the Xerox 8010 system at Xerox. A leading advocate called EMACS “the one true editor.”

Roberts¹ found that the overall performance time of display editors is only half that of line-oriented editors, though display editors are slower than line editors. The evidence supports the enthusiasm of display editor devotees. Furthermore, user automation evaluations consistently favor full-page display editors for secretarial and executive work.

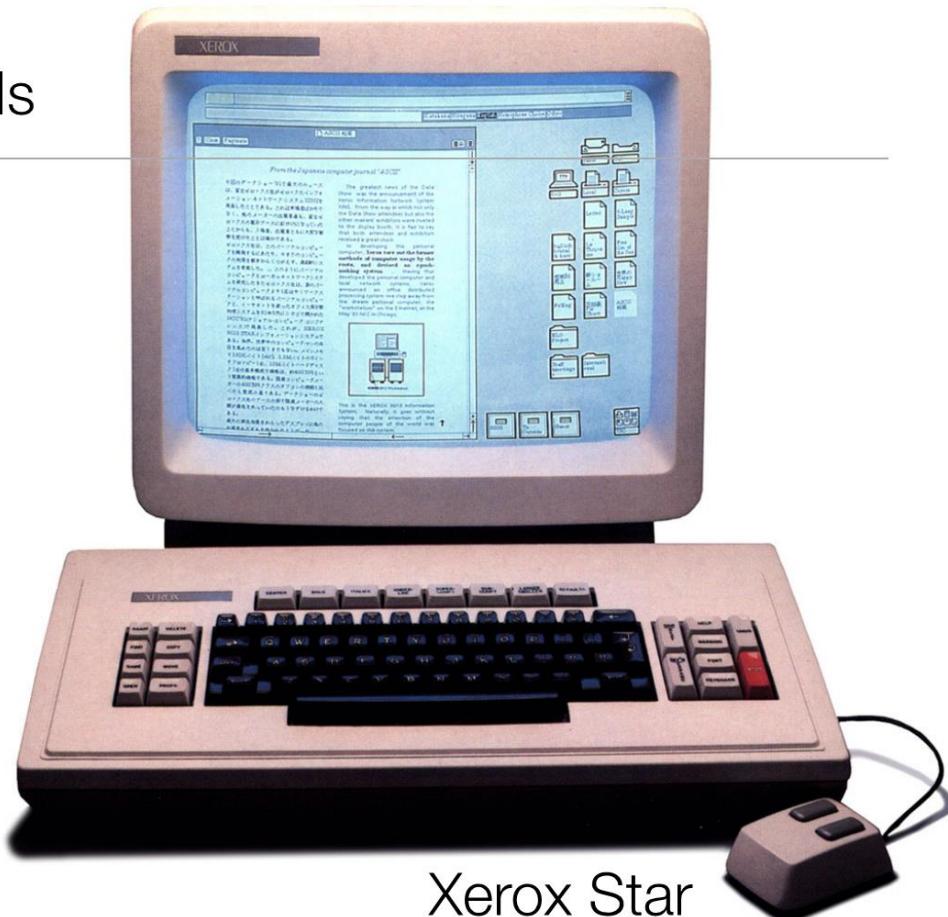
The advantages of display editors include

Display of a full 24 to 66 lines of text. This full display enables viewing each sentence in context and simplifies reading and scanning the document. By contrast, the

A portion of this article was derived from the author's keynote address at the NYU Symposium on User Interfaces, “The Future of Interactive Systems and the Emergence of Direct Manipulation,” published in *Proceedings of the Conference on Computer Systems, Y. Vallée, ed., Able Publishing Co., Norwood, N.J., 1983.*

August 1983 50(8) 51-55 0004-0735/83/08051-05\$01.00 © 1983 IEEE

1980
Commercial GUIs

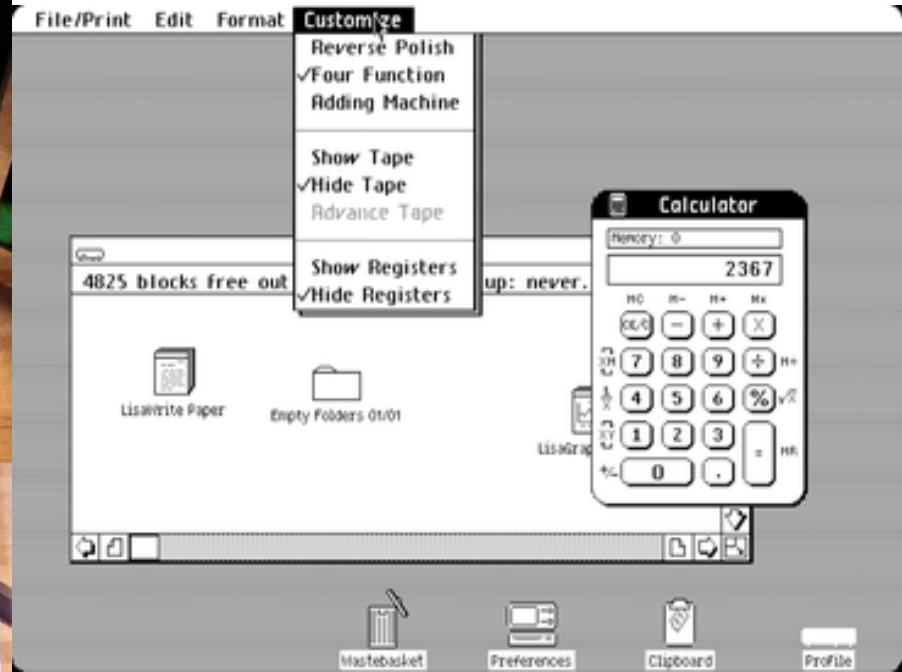


Xerox Star

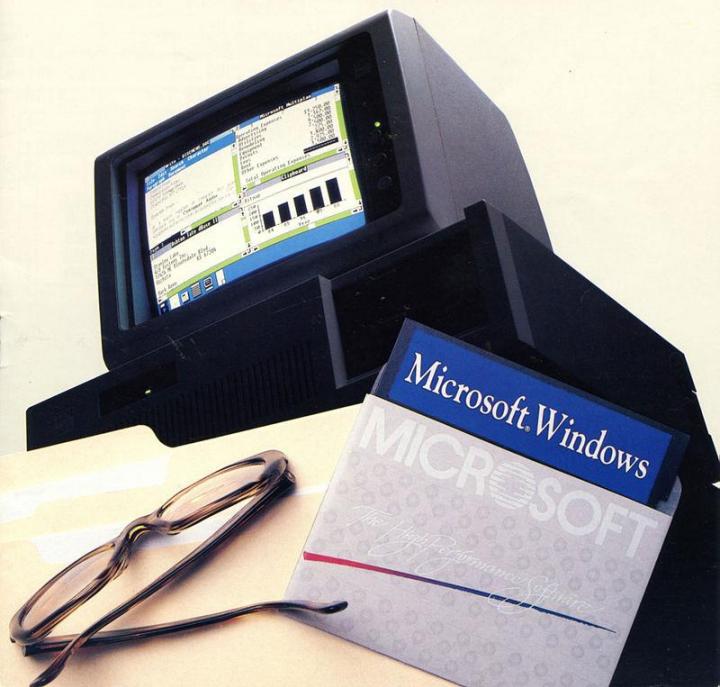
Jb Labrune, Jamie Zigelbaum, Hiroshi Ishii From. PreHistoric Interfaces to NearFuture Interactions



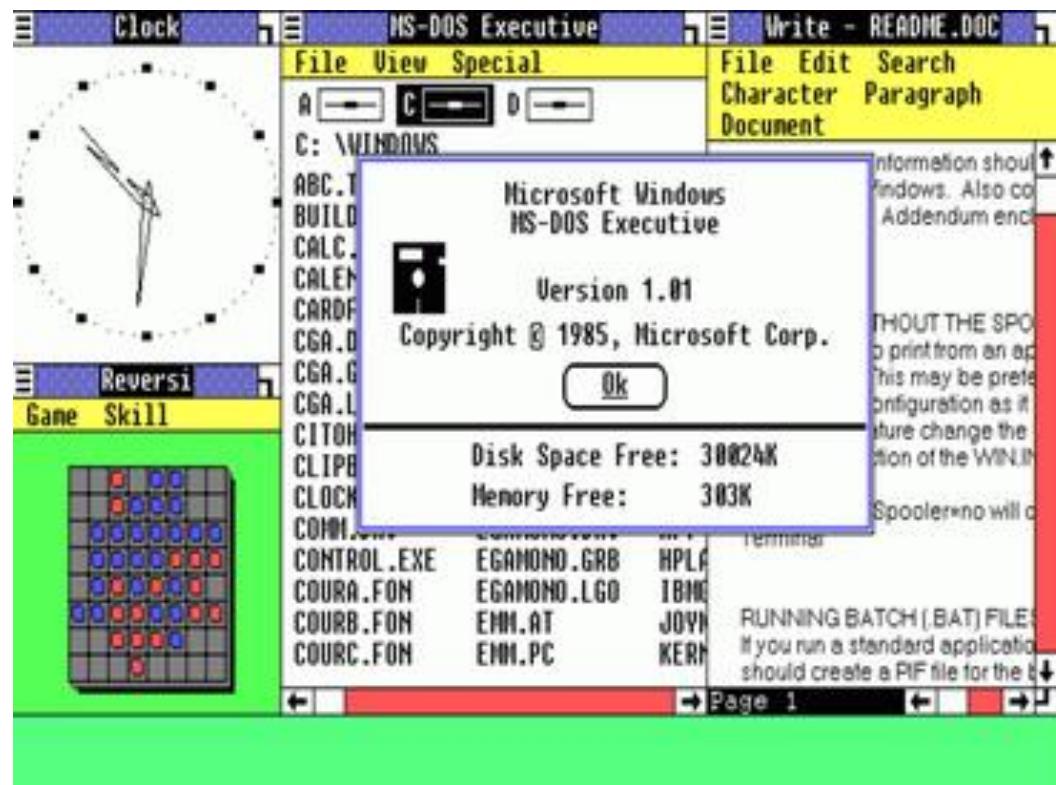
Apple Lisa, 1983



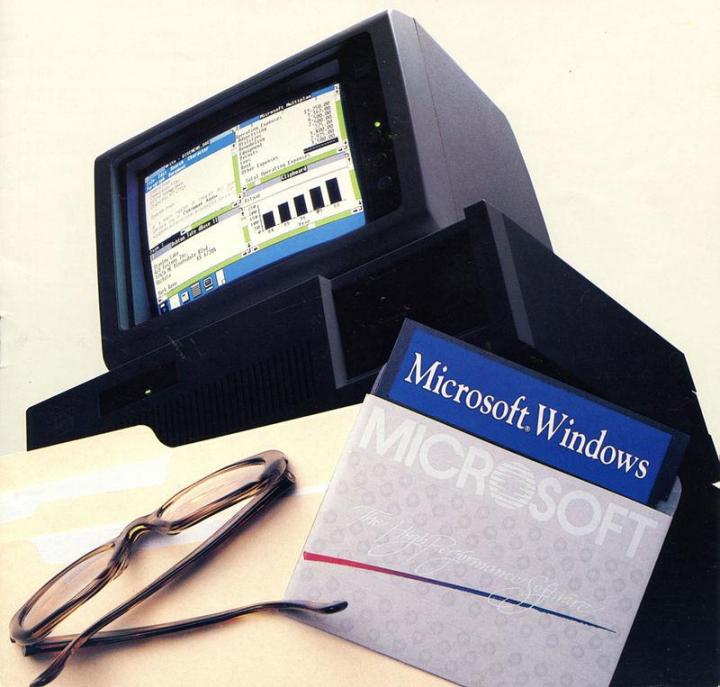
Microsoft Introduces Power Windows.



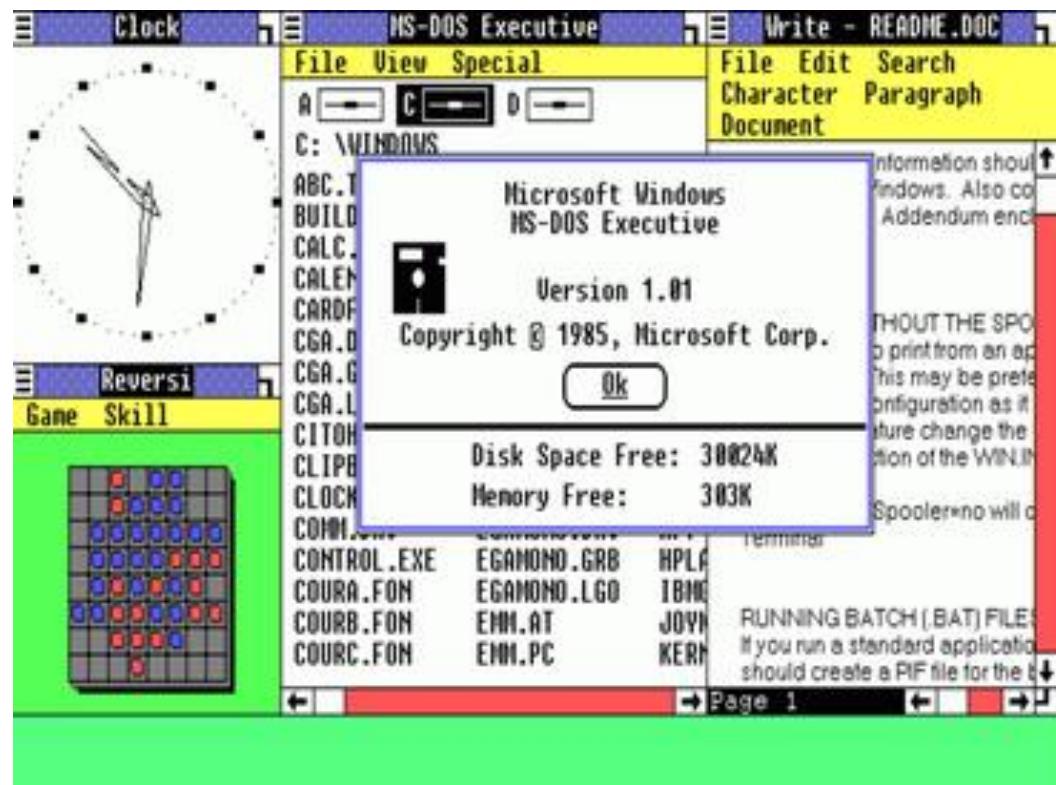
Windows 1.0, 1985



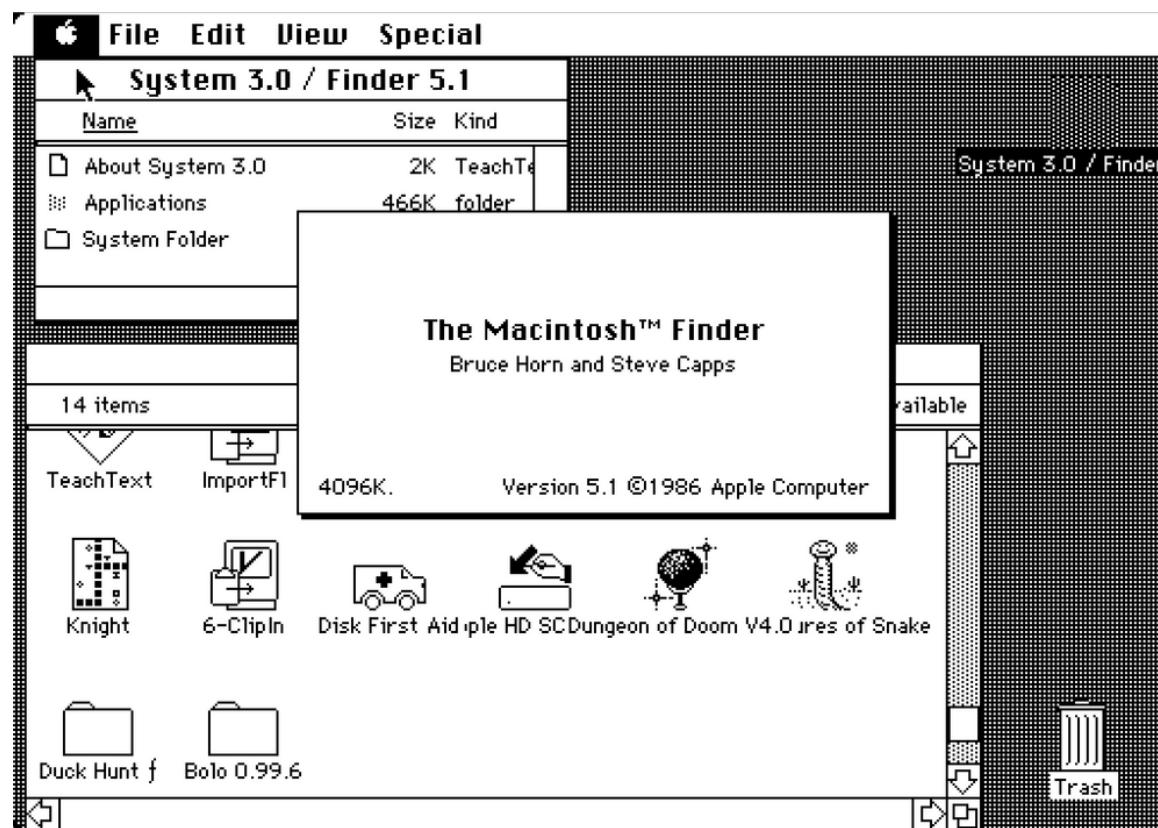
Microsoft Introduces Power Windows.



Windows 1.0, 1985

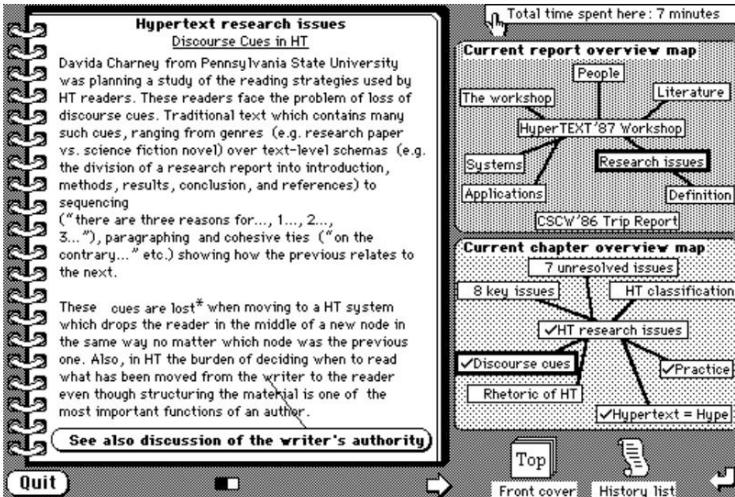


Apple System 3.0, 1986

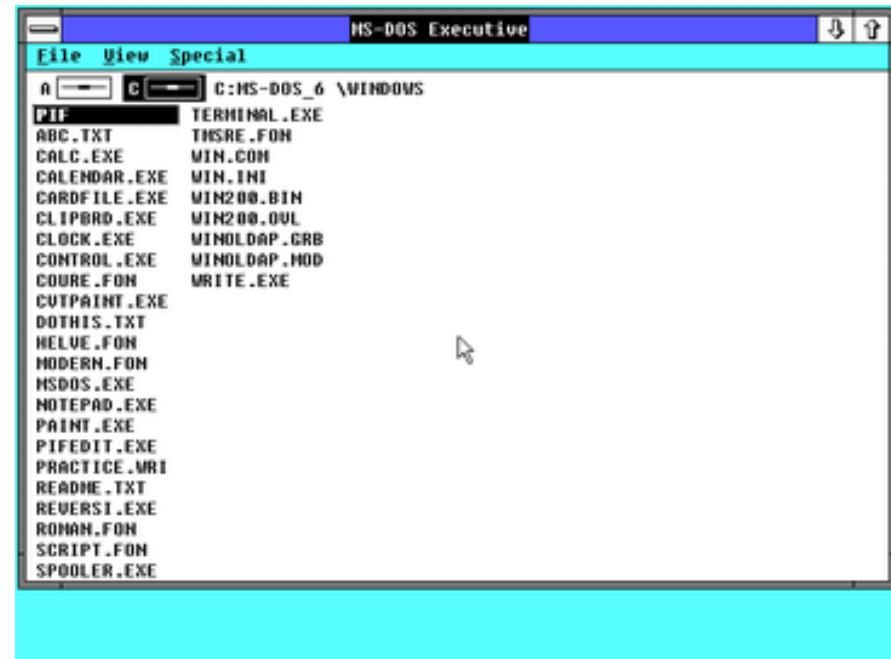
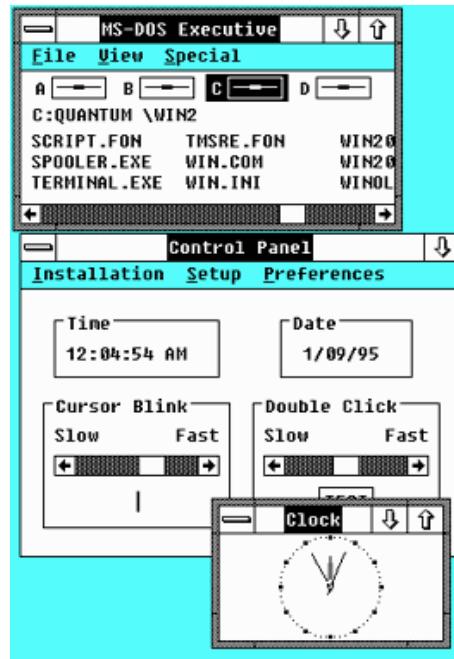


1987

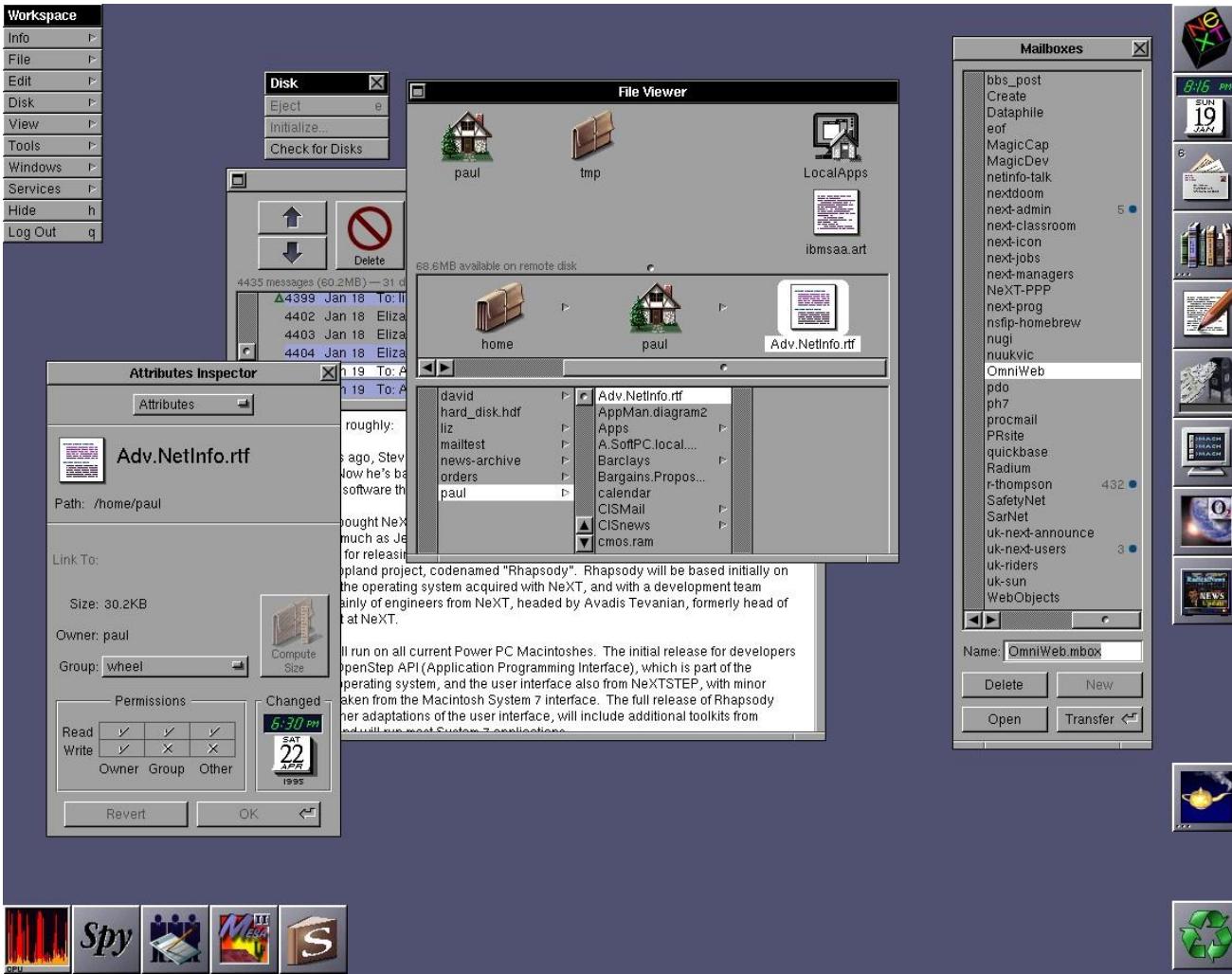
Hypercard to WWW



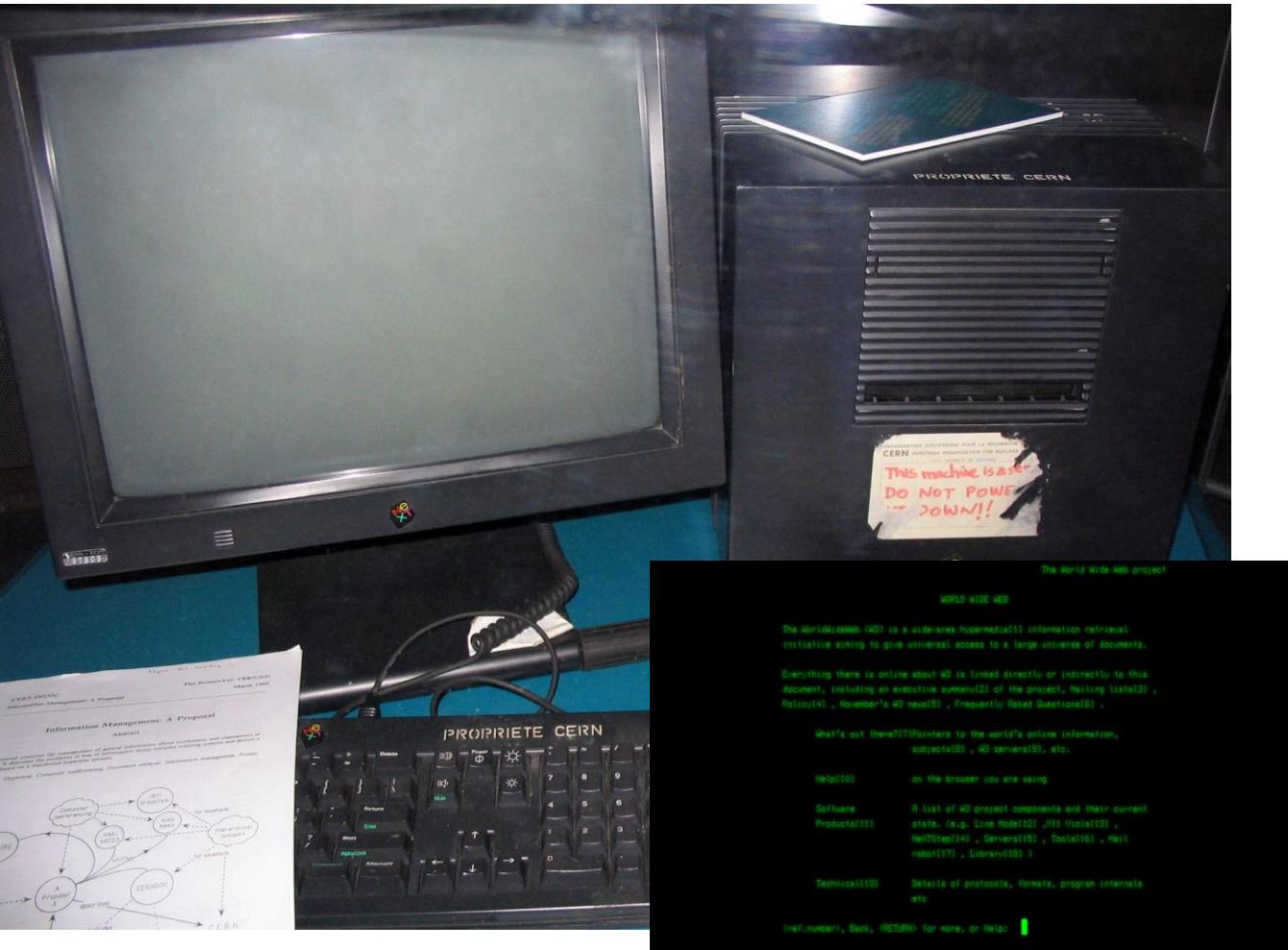
Windows 2.0, 1987



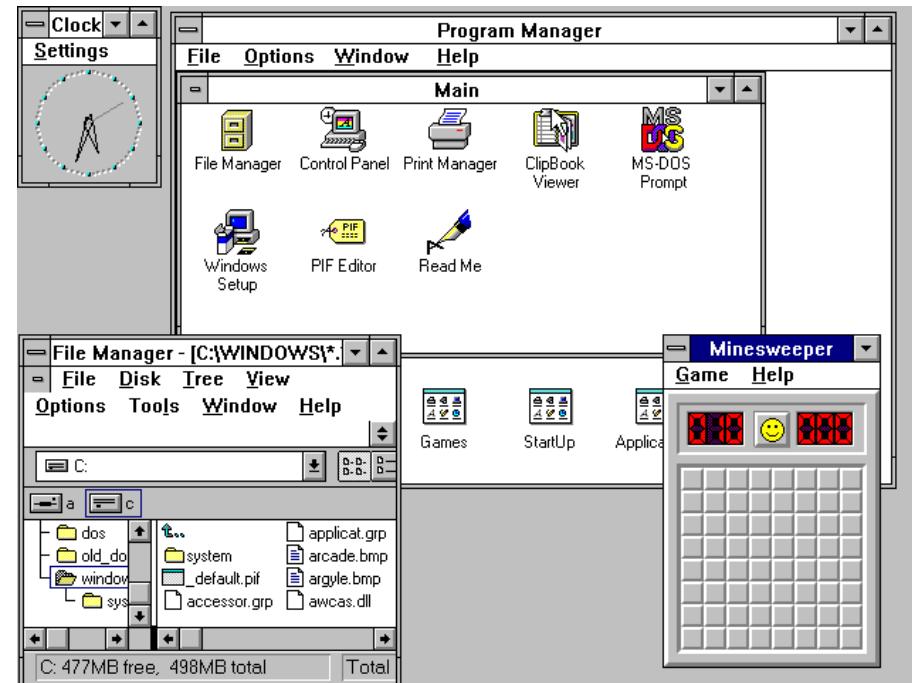
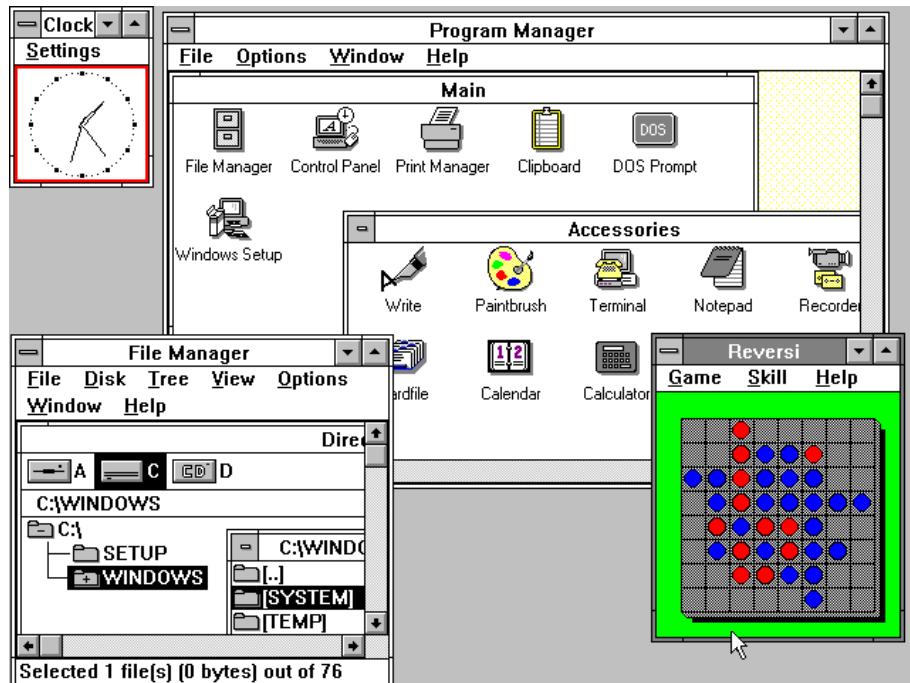
NeXTSTEP,
1989 =>
macOS, iOS,
iPadOS,
watchOS,
tvOS
pagrindas



This NeXT Computer used by Tim Berners-Lee at CERN became the world's first web server, 1990



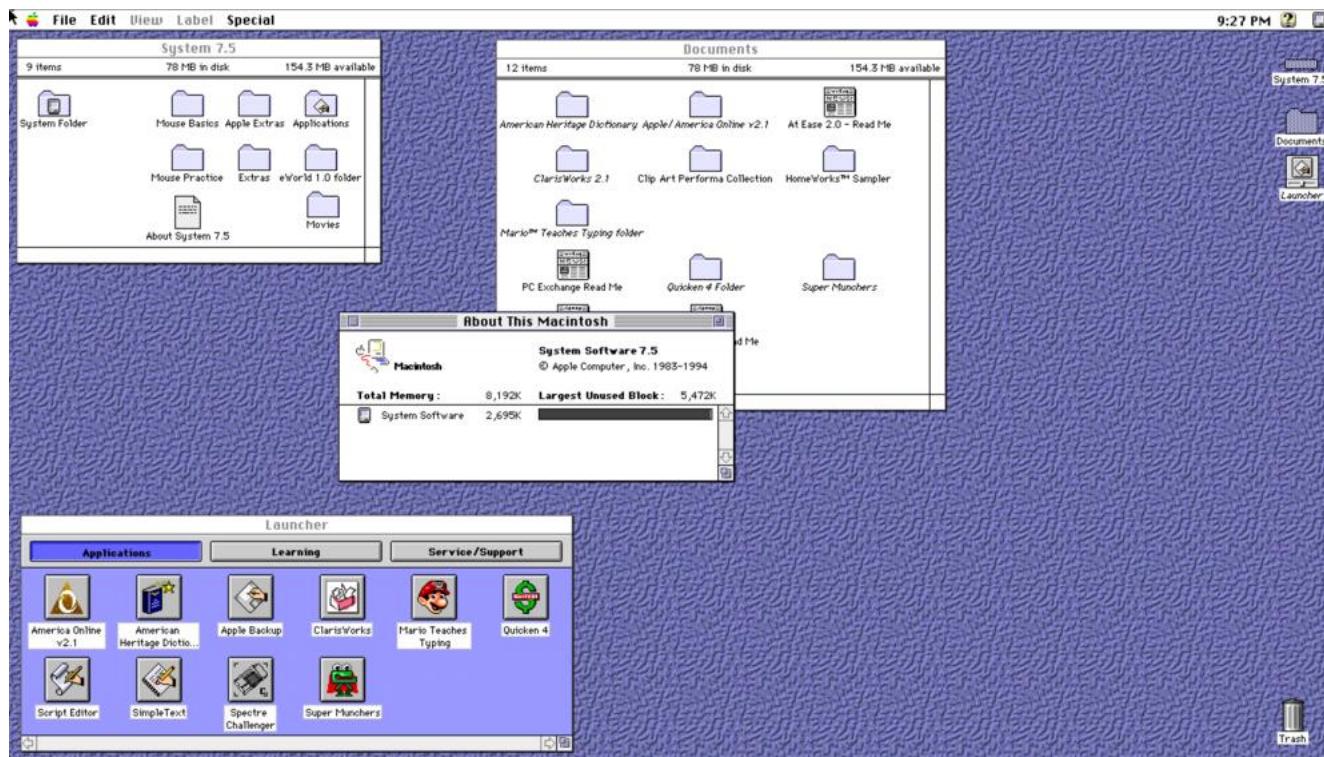
Windows 3.0, 1990; Windows 3.1x, 1992



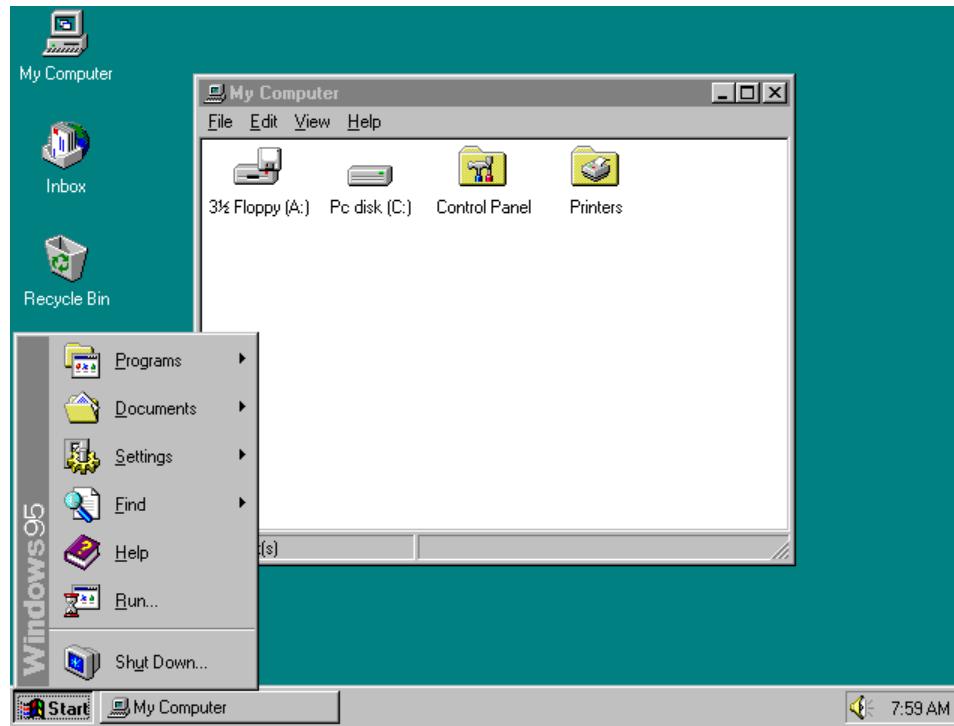
Apple System 7, 1991



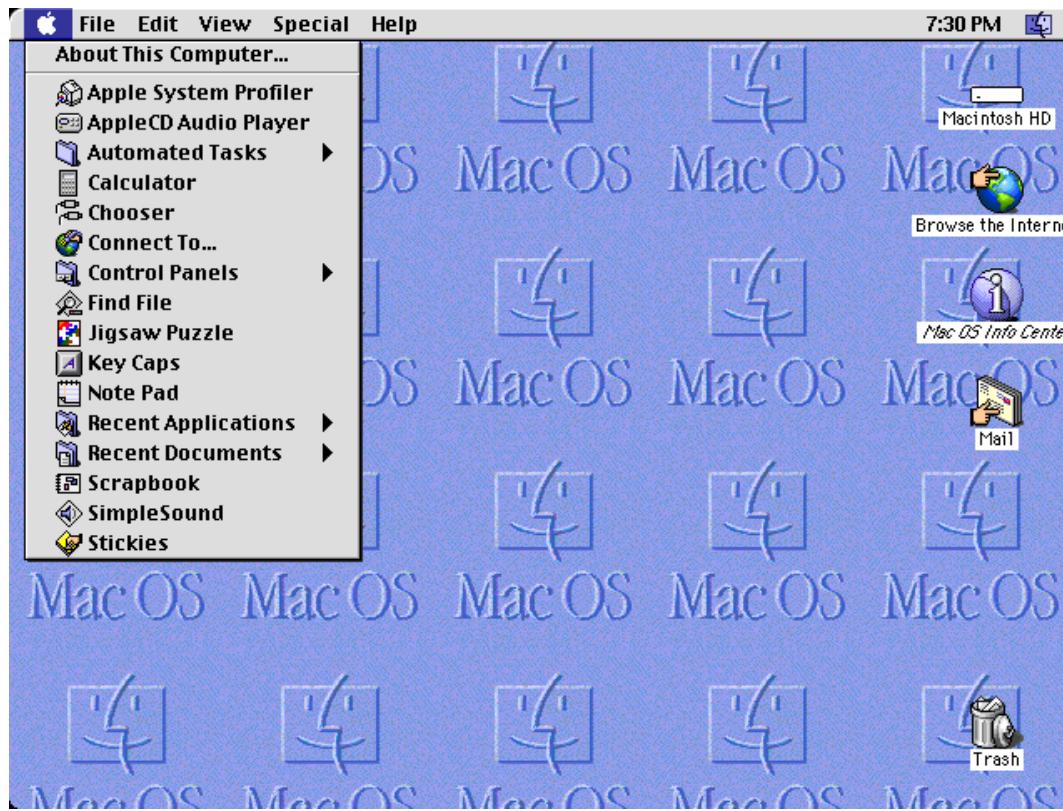
Apple System / Mac OS 7.5.x, 1994



Windows 95, 1995



Mac OS 8, 1997



MacOS X, 2000



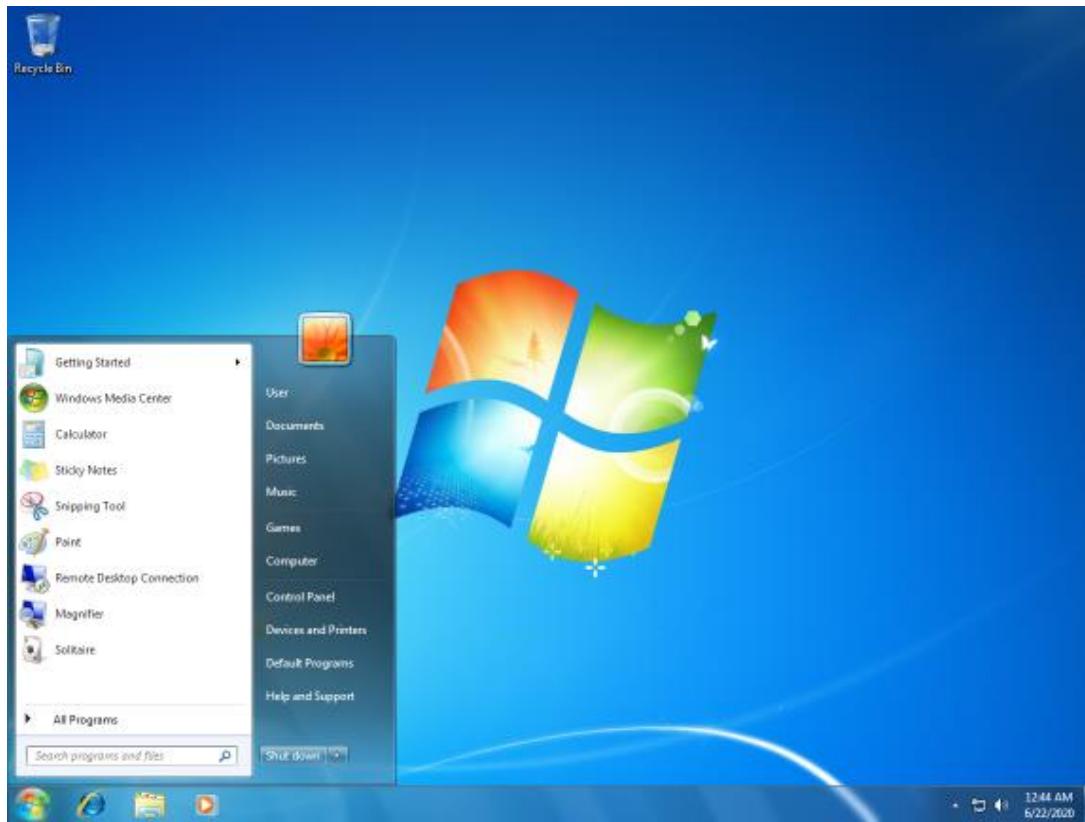
Windows XP, 2001



Windows Vista, 2006



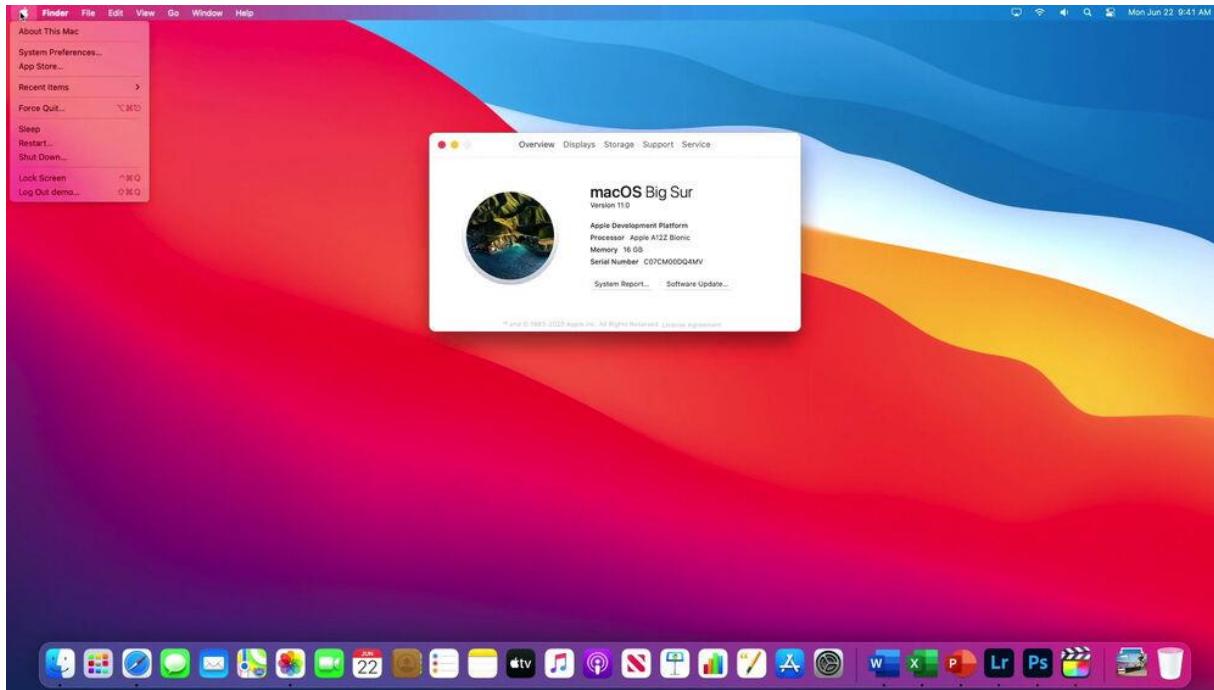
Windows 7, 2009



Windows 8, 2012

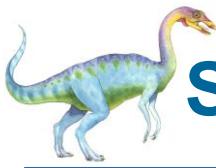


MacOS Big Sur, 2020



MacOS Big Sur, 2020





Sisteminiai iškvetimai (angl. System Calls)

- Tai programavimo sasaja servisams suteikiamą OS.
- Paprastai, parašyta aukšto lygio kalba (C ar C++).
- Dažniausiai pasiekiamą programų per aukšto lygio ***Application Programming Interface (API)***, negu per tiesioginius sisteminius iškvetimus.
- Trys dažniausiai paplitę API yra Win32 API Windows sistemose, POSIX API POSIX-paremtose sistemose (įskaitant praktiškai visas UNIX, Linux, ir Mac OS X versijas) ir Java API Java virtualioms mašinoms (JVM).

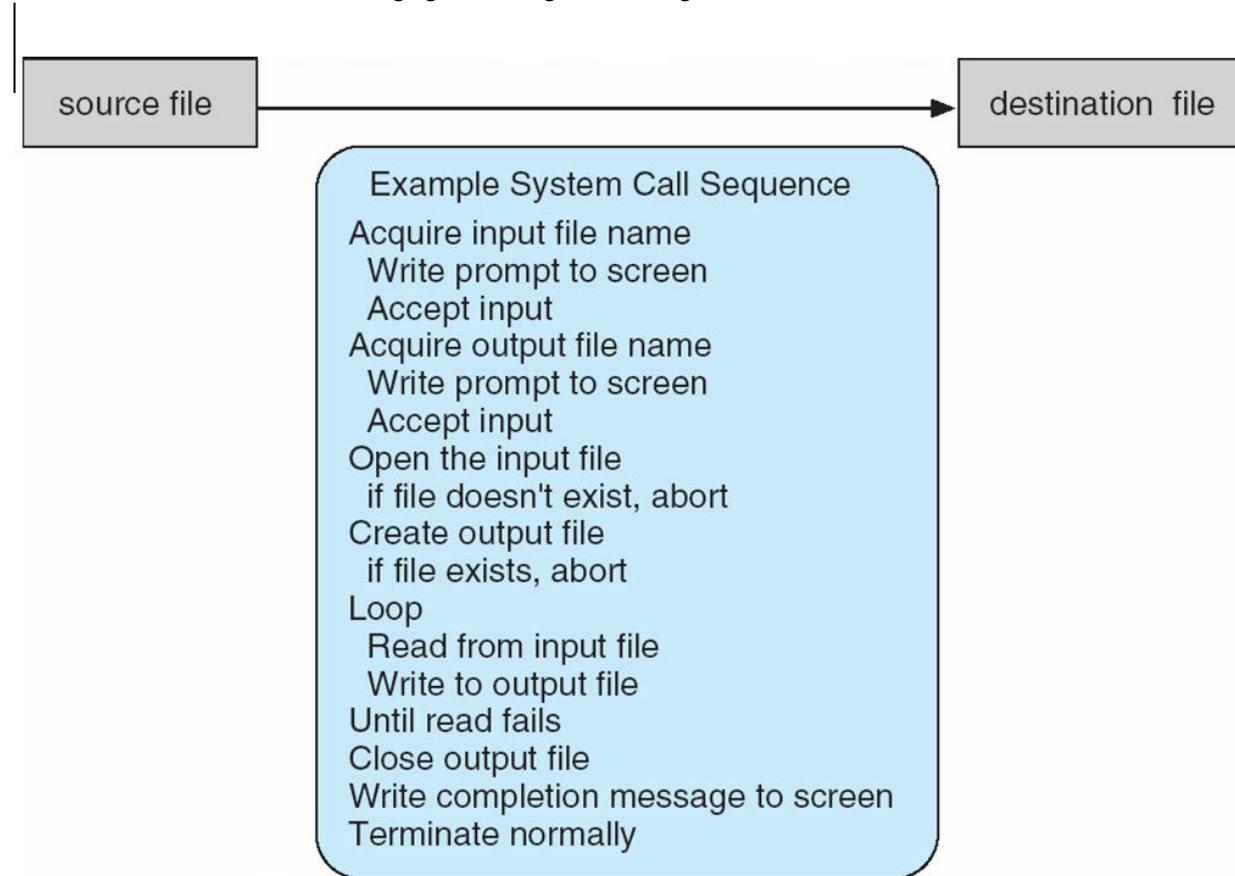
Pabrėžtina, kad sisteminų iškvetimų vardai, naudojami šiose paskaitose yra bendriniai





Sisteminių iškvietimų puz.

- Sisteminių iškvietimų seka, reikalinga nukopijuoti vieno failo turinį į kitą failą.





Standartinio API pavyzdys

EXAMPLE OF STANDARD API

As an example of a standard API, consider the `read()` function that is available in UNIX and Linux systems. The API for this function is obtained from the `man` page by invoking the command

```
man read
```

on the command line. A description of this API appears below:

```
#include <unistd.h>

ssize_t      read(int fd, void *buf, size_t count)
```

return value function name parameters

A program that uses the `read()` function must include the `unistd.h` header file, as this file defines the `ssize_t` and `size_t` data types (among other things). The parameters passed to `read()` are as follows:

- `int fd`—the file descriptor to be read
- `void *buf`—a buffer where the data will be read into
- `size_t count`—the maximum number of bytes to be read into the buffer

On a successful read, the number of bytes read is returned. A return value of 0 indicates end of file. If an error occurs, `read()` returns -1.





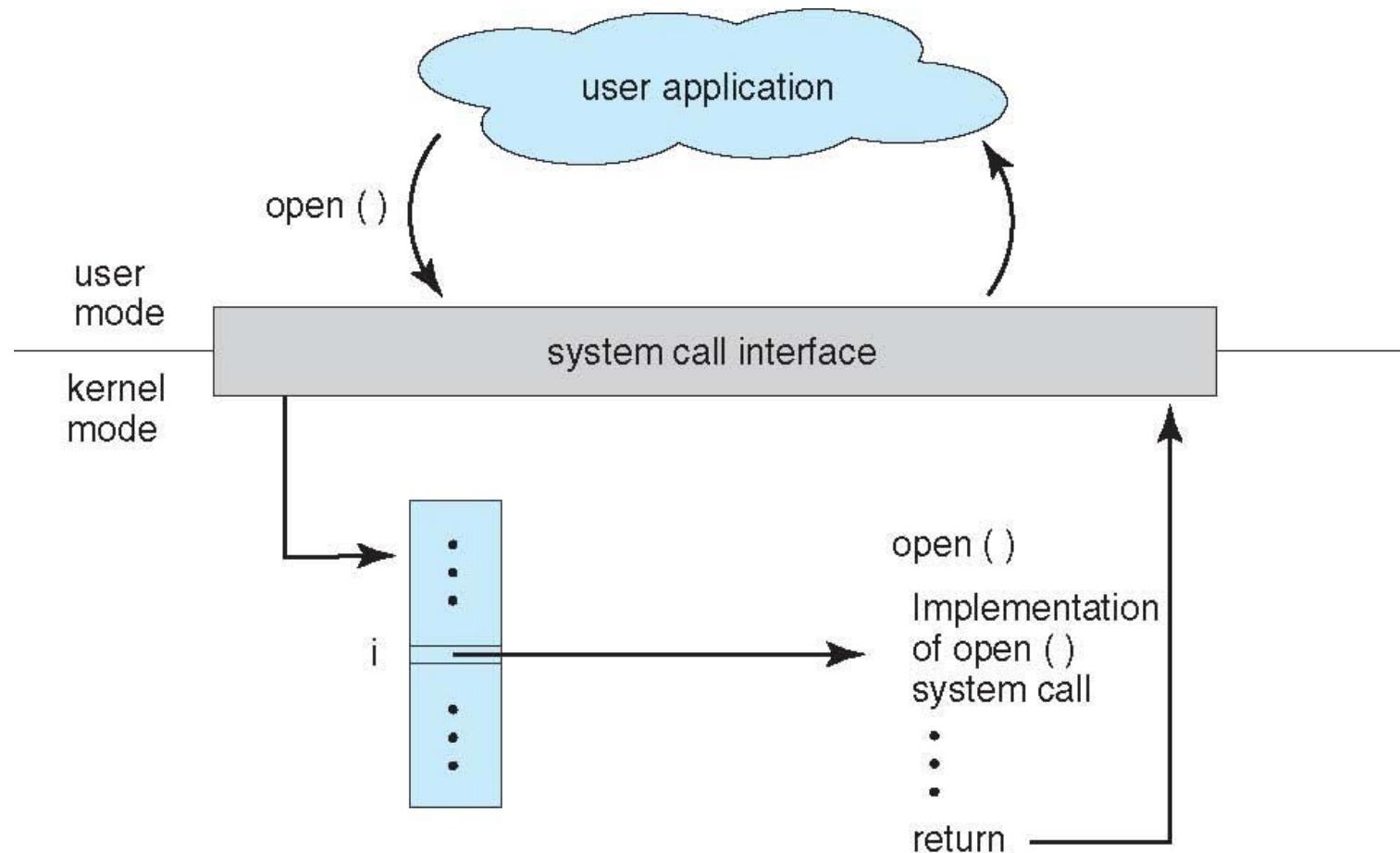
Sisteminių iškvietimų įgyvendinimas

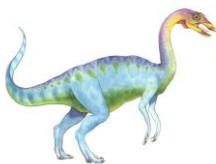
- Paprastai, yra skaičius susietas su kiekvienu sisteminių iškvietimu.
 - **System-call interface** saugo lentelę su tų skaičiu indeksais.
- Sisteminio iškvietimo sasaja iškviečia reikiama iškvietimą OS branduolyje ir grąžina sisteminio iškvietimo būseną ir reikiamas vertes.
- Iškvietėjui nereikia žinoti apie tai, kaip iškvietimas yra įgyvendintas.
 - Reikia tik suprasti API ir suprasti, ką OS atliks, kaip iškvietimo rezultataj.
 - Dauguma OS sasajos detalių yra paslėptos nuo programuotojo per API.
 - ▶ Valdomos run-time support library (funkcijų rinkinys, integruotas kompiliatoriuje)





API – Sisteminis iškvietas – OS santykis





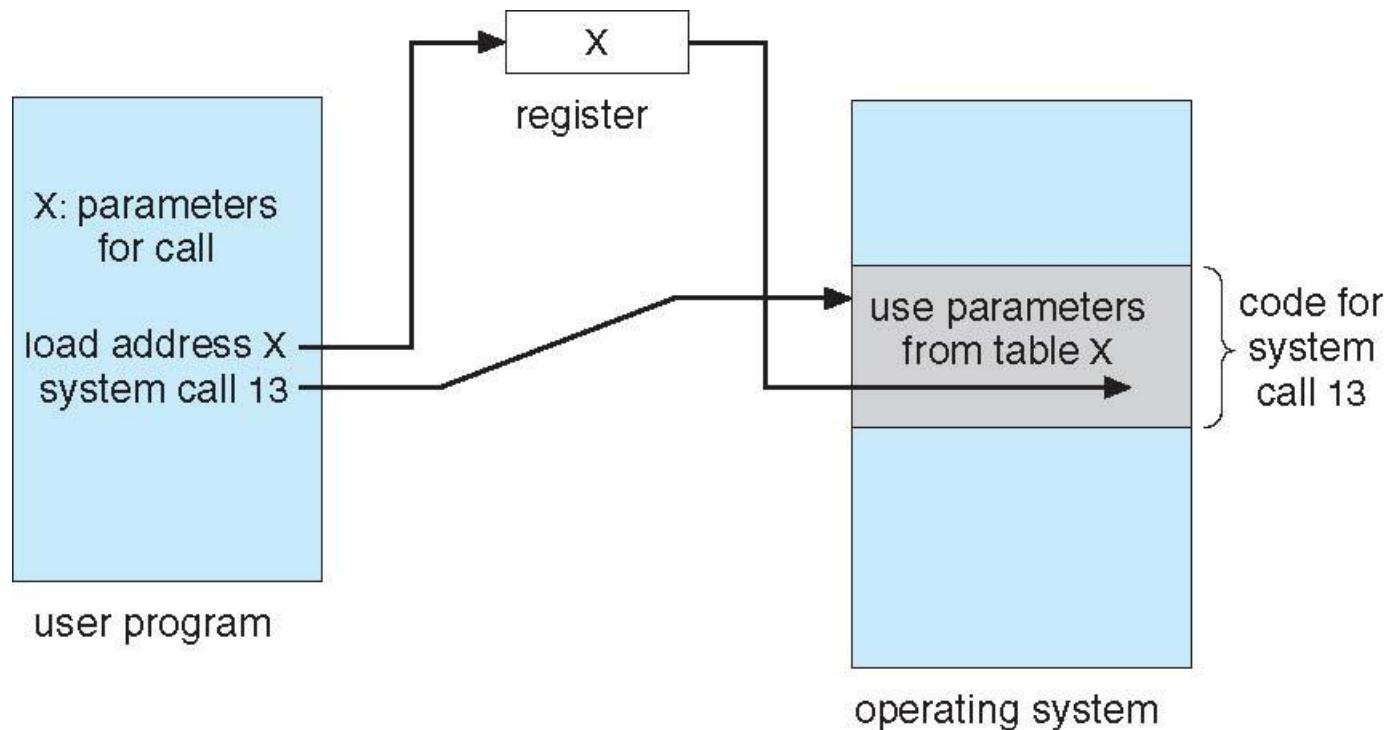
Sisteminio iškvietimo parametru perdavimas

- Dažnai, yra reikalinga detalesnė informacija, nei vien reikiama sisteminio iškvietimo identifikavimas.
 - Konkretus tipas ir informacijos kiekis skiriasi, priklausomai nuo OS ir iškvietimo.
- Trys pagrindiniai metodai, naudojami perduoti parametrus OS
 - Paprasčiausias: perduoti parametrus registruose.
 - ▶ Kai kuriais atvejais, gali būti daugiau parametru nei vien registratori
 - Parametrai saugomi blokuose ar lentelėse atmintyje, o bloko adresas perduodamas, kaip parametras registre.
 - ▶ Šis metodas taikomas Linux ir Solaris
 - Parametrai yra padedami, arba stumiami (angl. *pushed*), į steką (angl. *stack*) programos ir išmušami (angl. *popped*) iš steko operacinės sistemas.
 - Blokų ir steko metodai neriboja perduodamų parametru skaičiaus ar ilgio.





Parametru perdavimas per lentelę





Sisteminių iškvietimų tipai

- Procesų valdymo
 - create process, terminate process
 - end, abort
 - load, execute
 - get process attributes, set process attributes
 - wait for time
 - wait event, signal event
 - allocate and free memory
 - Dump memory if error
 - **Debugger** for determining **bugs, single step** execution
 - **Locks** for managing access to shared data between processes





Sisteminių iškvietimų tipai

- Failų valdymo
 - create file, delete file
 - open, close file
 - read, write, reposition
 - get and set file attributes
- Įrenginių valdymo
 - request device, release device
 - read, write, reposition
 - get device attributes, set device attributes
 - logically attach or detach devices





Sisteminių iškvietimų tipai (tęs.)

- Informacijos valdymo
 - get time or date, set time or date
 - get system data, set system data
 - get and set process, file, or device attributes
- Komunikacijos
 - create, delete communication connection
 - send, receive messages if **message passing model** to **host name** or **process name**
 - ▶ From **client** to **server**
 - **Shared-memory model** create and gain access to memory regions
 - transfer status information
 - attach and detach remote devices





Sisteminių iškvietimų tipai (tęs.)

- Apsaugos
 - Control access to resources
 - Get and set permissions
 - Allow and deny user access





Windows ir Unix sisteminėmis iškvietimų pavyzdžiai

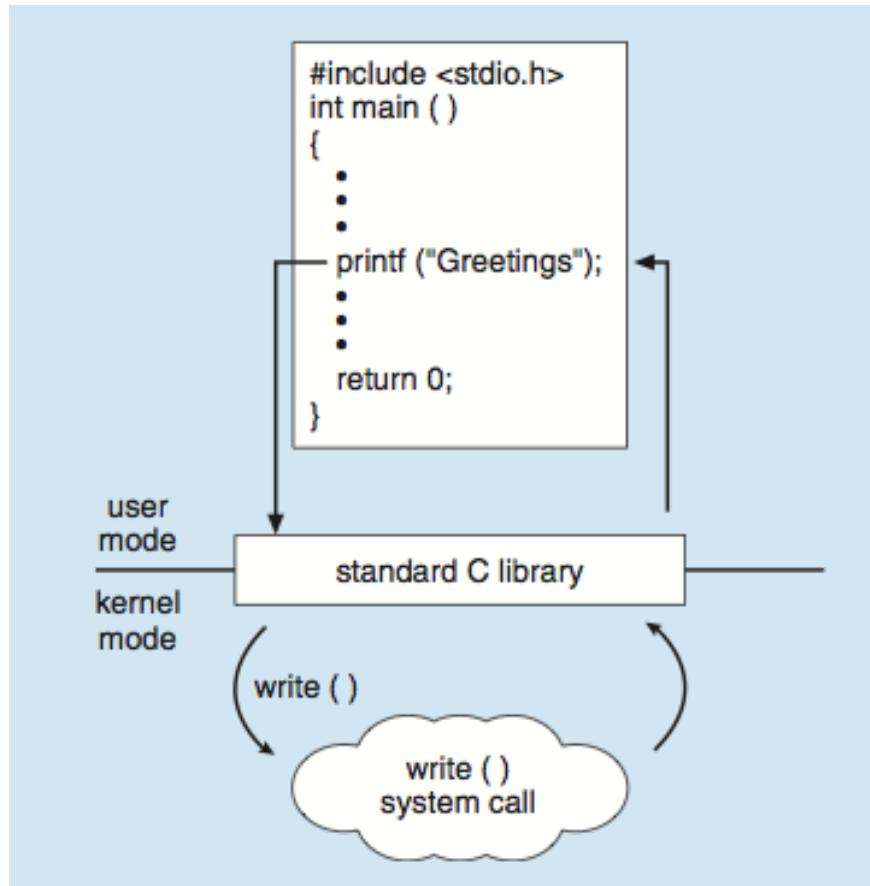
	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

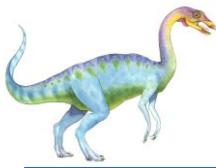




Standartinės C bibliotekos pavyzdys

- C programa, iškviečianti printf() sisteminį iškvietimą, kuris iškviečia write() sisteminį iškvietimą.

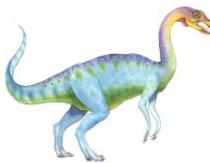




Sisteminės programos

- Sisteminės programos suteikia patogią aplinką programų kūrimui ir vykdymui. Jos gali būti suskirstytos į:
 - Failų manipuliavimo
 - Būsenos informacijos
 - Programavimo kalbų palaikymo
 - Programų paleidimo ir vykdymo
 - Komunikacijos
 - Foninių servisų
 - Taikomujų programų
- Daugelio vartotojų požiūris į OS yra aprašomas, kaip sisteminės programos, ne kaip konkretūs sisteminiai iškvietimai.

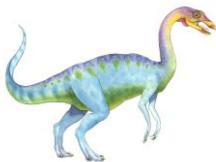




Sisteminės programos

- Suteikia patogią aplinką programų kūrimui ir vykdymui.
 - Kai kurios iš jų yra paprastos vartotojo sąsajos prie sisteminiu iškvietimui, kitos – ženkliai sudėtingesnės.
- **Failų valdymas** – sukurti, ištrinti, kopijuoti, pervadinti, spausdinti, išvesti sąrašą ir bendrai atlikti manipuliacijas su failais ir katalogais.
- **Būsenos informacija**
 - Kai kurios gauna informaciją iš sistemos - datą, laiką, prieinamos atminties vietą, kietojo disko vietą, vartotojų skaičių.
 - Kitos suteikia detalesnę informaciją ir žurnalus apie spartą ir derinimo informaciją.
 - Paprastai, šios programos suformuoja ir išveda informaciją terminale ar kituose išvesties įrenginiuose.
 - Kai kuriose sistemoje įgyvendinti **registratorai** – naudojami saugoti ir gauti konfigūracijos informacijai.

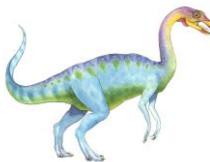




Sisteminės programos (tęs.)

- **Failų modifikavimas**
 - Tekstų redaktoriai naudojami kurti ir modifikuoti failus
 - Naudojamos specialios komandos ieškoti failo turinio ar atlikti teksto transformavimą.
- **Programavimo kalbų palaikymas** - kompiliatoriai, asembleriai, debugeriai ir interpretatoriai
- **Programų paleidimas ir vykdymas** – absolютiniai paleidėjai (angl. absolute loaders), perkeliami paleidėjai (angl. relocatable loaders), derinimo sistemos aukštesnio lygio ir mašininėms kalboms.
- **Kominikacijos** – suteikia mechanizmus kurti virtualiems ryšiams tarp procesų, vartotojų ir kompiuterinių sistemų.
 - Leidžia vartotojams siųst žinutes į kito vartotojo ekraną, naršyti interneto puslapius, siųsti e. laiškus, prisijungti nuotoliniu būdu, perduoti failus iš vieno kompiuterio į kitą.





Sisteminės programos (tęs.)

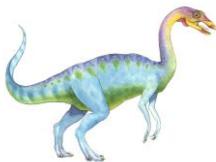
■ Foniniai servisai

- Paleidžiami įkrovos metu
 - ▶ Kai kurie tik sistemos paleidimui, po to sustabdomi.
 - ▶ Kai kurie veikia nuo sistemos įkrovos iki išjungimo.
- Suteikia paslaugas tokias, kaip diskų tikrinimas, procesų tvarkaraščių sudarymas, klaidų žurnalizavimas, spausdinimas.
- Veikia vartotojo kontekste, o ne branduolio.
- Žinomi, kaip **services**, **subsystems**, **daemons**

■ Taikomosios programos

- Nėra susietos su sistema
- Vykdamos vartotojų
- Neskaitomas OS dalimi
- Paleidžiamos iš komandų eilutės, pelės paspaudimu, piršto palietimu.





OS projektavimas ir įgyvendinimas

- Nėra vieningo sprendimo, tačiau kai kurie sprendimai pasirodė sėkmingi.
- Vidinė OS struktūra gali ženkliai skirtis.
- Projektavimas pradedamas aprašant tikslus ir specifikacijas.
- Įtakojamas techninės įrangos pasirinkimu, sistemos tipu.
- **Vartotojo** tikslai ir **sistemos** tikslai:
 - Vartotojo tikslai – OS turi būti patogi naudoti, lengva išmokti, patikima, saugi ir greita.
 - Sistemos tikslai – OS turi būti lengvai suprojektuojama, įgyvendinama ir prižiūrima, turi būti lanksti, patikima, be klaidų ir efektyvi.





OS projektavimas ir įgyvendinimas (tęs.)

- Svarbu atskirti principus:
Strategija: *Kas bus daroma?*
Mechanizmas: *Kaip tą padaryti?*
- Mechanizmai nurodo kaip kažką padaryti, strategijos – kas bus padaryta.
- OS specifikavimas ir projektavimas yra labai kūrybinga **programų inžinerijos** (angl. *software engineering*) sritis.





Igyvendinimas

- Daug variacijų
 - Ankstyvosios OS rašomos asemblerio kalba.
 - Vėliau – sisteminėmis programavimo kalbomis, kaip Algol, PL/1.
 - Dabar - C, C++.
- Dažniausiai naudojamas kalbų mišinys:
 - Žemiausio lygmens kodas – asembleriu.
 - Pagrindinis kodas – C.
 - Sisteminės programos - C, C++, skriptų kalbomis, kaip PERL, Python, shell skriptais.
- Kuo aukštesnio lygmens kalba – tuo lengviau perkelti (angl. **port**) į kitą techninę įrangą.
 - Tačiau tuo lėčiau.
- **Emuliacija** leidžia OS veikti svetimoje (angl. non-native) techninėje įrangoje.





OS struktūra

- Bendrosios paskirties OS yra labai didelė programa.
- Daug skirtingų būdu struktūrizuoti:
 - Paprasta struktūra – MS-DOS
 - Sudėtingesnė- UNIX
 - Sluoksniuota – abstrakcija
 - Microkernel

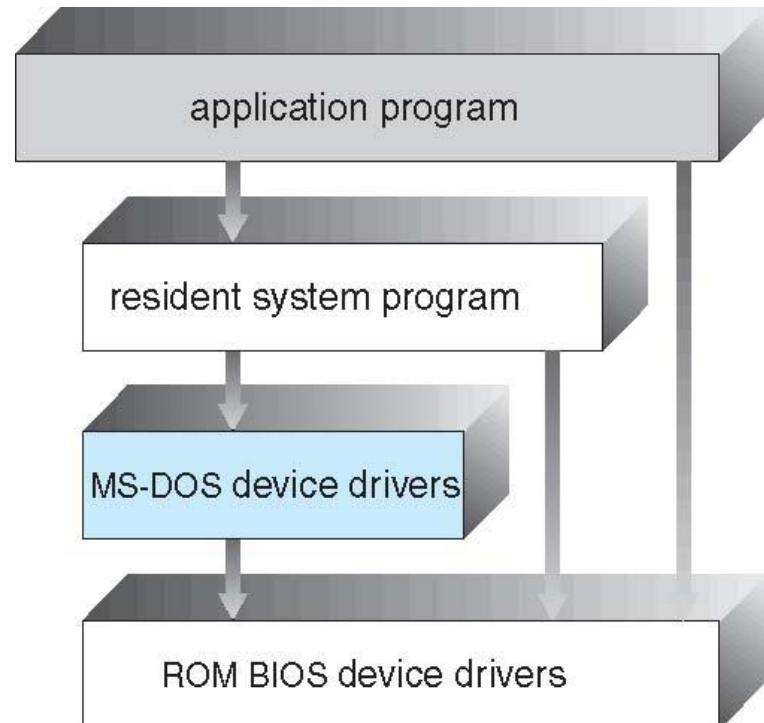




Paprasta struktūra - MS-DOS

- MS-DOS – parašyta, kad suteikti daugiausiai funkcionalumo, užimant mažiausiai vietos.
 - Neskirstoma į modulius
 - Nors MS-DOS turi tam tikrą struktūrą, tačiau jos sąsajos ir funkcionalumo lygiai nėra gerai atskirti.

```
Starting MS-DOS...  
C:\>_
```





MS-DOS

Pavadinimas	Maksimalus kietojo diskų dydis	Standartiškai palaikoma failinė sistema	Standartinis 3.5" diskelių palaikymas	Standartinis 5.25" diskelių palaikymas	Integruota diskų suspaudimo galimybė	Ilgų failo vardu standartinis palaikymas
MS-DOS 1.1	n/a	FAT12	n/a	320 kB (dvipusis)	Ne	Ne
MS-DOS 2.0	10 MB	FAT12	n/a	360 kB	Ne	Ne
MS-DOS 3.0	32 MB	FAT12	n/a	360 kB, 1.2 MB	Ne	Ne
MS-DOS 3.2	32 MB	FAT12	720 kB	360 kB, 1.2 MB	Ne	Ne
MS-DOS 3.3	32 MB	FAT12	720 kB, 1.44 MB	360 kB, 1.2 MB	Ne	Ne
MS-DOS 4.0	2 GB	FAT12, FAT16	720 kB, 1.44 MB	360 kB, 1.2 MB	Ne	Ne
MS-DOS 5.0	2 GB	FAT12, FAT16	720 kB, 1.44 MB, 2.88 MB	360 kB, 1.2 MB	Ne	Ne
MS-DOS 6.0	2 GB	FAT12, FAT16	720 kB, 1.44 MB, 2.88 MB	360 kB, 1.2 MB	Dviguba erdvė	Ne
MS-DOS 6.22	2 GB	FAT12, FAT16	720 kB, 1.44 MB, 2.88 MB	360 kB, 1.2 MB	Disko erdvė	Ne
MS-DOS 7.0 (Windows 95A)	2 GB	FAT12, FAT16	720 kB, 1.44 MB, 2.88 MB	360 kB, 1.2 MB	Disko erdvė	Ne (DOSLFN)
MS-DOS 7.1x (Windows 95B/OSR2, 95C/OSR2.5, 98, and 98SE)	124.55 GB (su FAT32)	FAT12, FAT16, FAT32	720 kB, 1.44 MB, 2.88 MB	360 kB, 1.2 MB	Disko erdvė Windows 95 versijai, Ne Windows 98 versijai	Ne (DOSLFN)
MS-DOS 8.0 (Windows Me)	124.55 GB (su FAT32)	FAT12, FAT16, FAT32	720 kB, 1.44 MB, 2.88 MB	360 kB, 1.2 MB	Ne	Ne (DOSLFN)



Sudėtinga struktūra- UNIX

UNIX – dėl riboto techninės įrangos funkcionalumo, pradinė UNIX OS turėjo ribotą struktūrą. UNIX OS susideda iš dviejų atskirų dalių:

- Sisteminių programų;
- Branduolio, kuris:
 - ▶ Susideda iš visko žemiau sisteminio iškvietimo sąsajos ir virš fizinės įrangos.
 - ▶ Suteikia failų sistemą, CPU tvarkaraščius, atminties valdymą ir kitas pagrindines OS funkcijas.

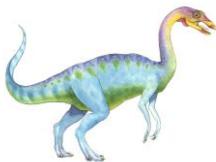




Sudėtinga struktūra- UNIX

- Iš kitų OS UNIX išsiskiria visų pirma failų sistema bei procesų valdymu.
- Daugelj UNIX bruožų yra perėmusios ir kitos OS, pvz., DOS ir Microsoft Windows naudojama katalogų sistema yra perimta iš UNIX, o OpenVMS, nors turi visiškai kitas šaknis, yra net sertifikuota kaip UNIX.
- Vienas iš neįprastų UNIX bruožų – bendra (nerašyta) ideologija, teigianti, kad OS turi būti patogi ir aiški specialistams, tuo tarpu eiliniai vartotojai, norėdami naudotis šiomis sistemomis, turi mokytis, bet nereikalauti sistemos profanavimo.

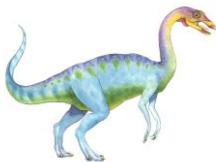




Sudėtinga struktūra- UNIX

- Failų sistema UNIX tipo OS abstrahuoja ne tik prijimą prie duomenų, bet ir daugelį OS funkcijų bei įrenginių, todėl didžiajų dalį OS leidžiamų veiksmų galima atlikti naudojantis standartinėmis teksto apdirbimo programomis.
- Naujesnėse UNIX versijose per failų sistemą galima pasiekti ir duomenis, susijusius su pačios OS veikimu, dirbančiais procesais ir t. t.
- Pvz., prieikus, visą disko turinį (įskaitant jo failų sistemą) galima nuskaityti, atidarius failą /dev/fd0, o garsinį dokumentą išklausyti, jį nukopijavus į /dev/dsp.
- Daugelį sistemos parametru galima skaityti ir reguliuoti, naudojantis failais, esančiais /proc/ kataloge.

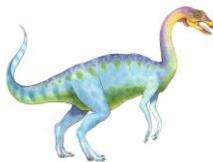




Sudėtinga struktūra- UNIX

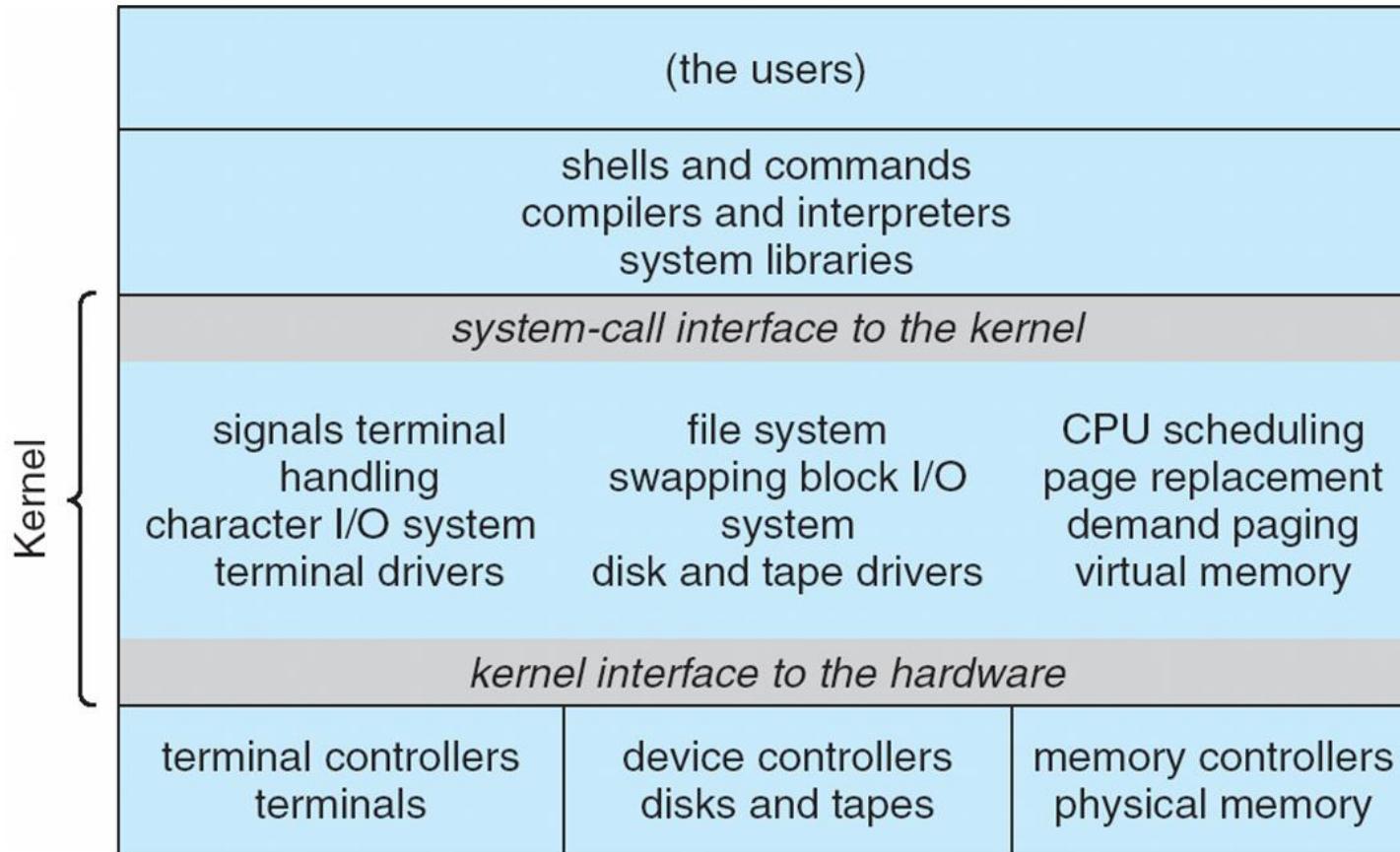
- UNIX turi gana netipišką procesų paleidimo metodą, nebūdingą daugumai kitų OS:
 - fork() – funkcija sukuria einamojo proceso kopiją, kuri nuo motininio proceso skiriasi tik identifikatoriumi. Skirtingai nuo kitų C kalbai būdingų funkcijų, ši grąžina du rezultatus: vieną – į motininį procesą (sukurto proceso identifikatorių), kitą – į dukterinį procesą (nulį).
 - exec() – negrąžina jokio rezultato, bet einamajį procesą pakeičia į kitą, naujai sukurtą iš vykdomojo failo, kuris nurodytas šiai funkcijai.
 - wait() – laukia, kol fork() sukurtas procesas pasibaigs, naudojant exit() funkciją, bei grąžina rezultatą, perduotą exit() funkcijai.
 - exit() – baigia procesą ir grąžina rezultatą motininiam procesui.





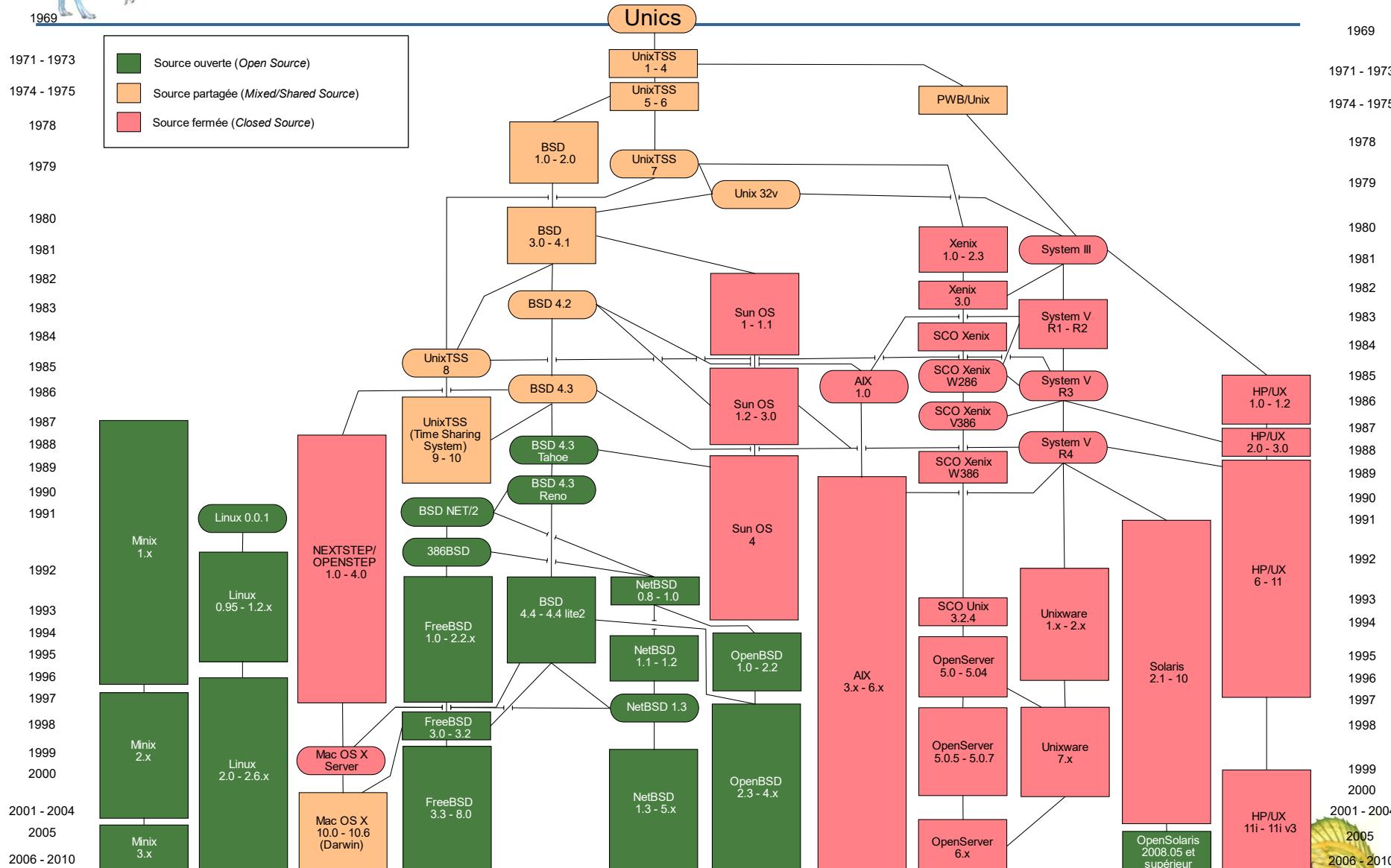
Tradicinė UNIX sistemos struktūra

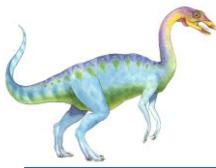
Paprasta, tačiau ne visiškai sluoksniuota





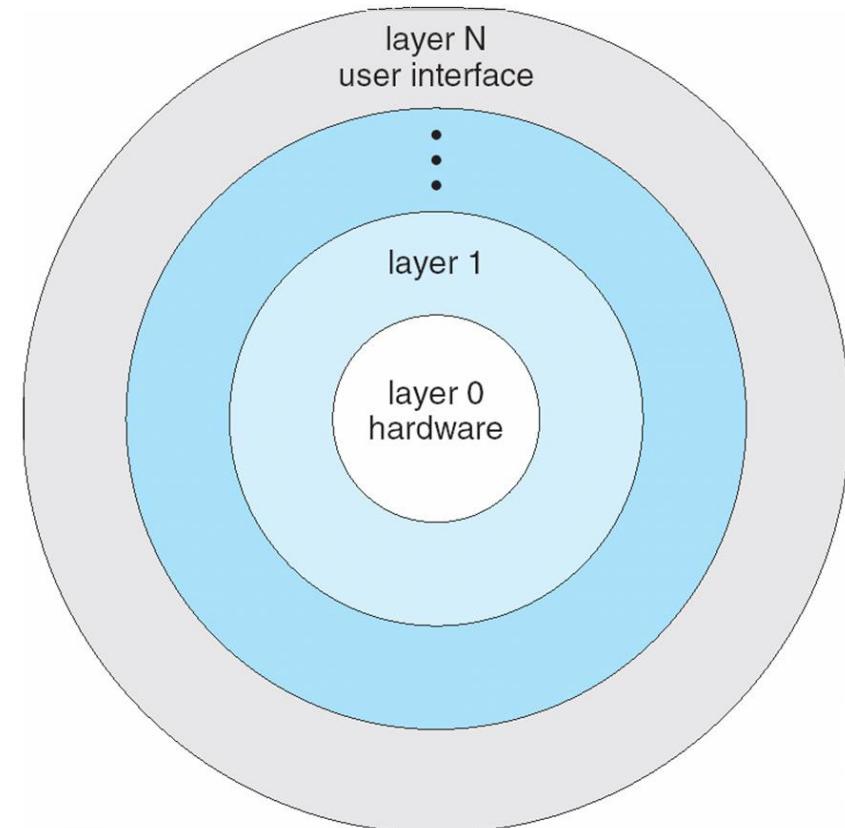
UNIX

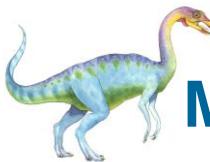




Sluoksniuotas požiūris

- OS padalinta į tam tikrą skaičių sluoksnį, kurių kiekvienas uždėtas ant žemesniųjų. Žemiausias sluoksnis (layer 0) yra techninė įranga; aukščiausias (layer N) – vartotojo sąsaja.
- Pagal modulumą, sluoksniai sudaromi taip, kad kiekvienas sluoksnis naudoja tik žemesniojo sluoksnio funkcijas ir servisus.





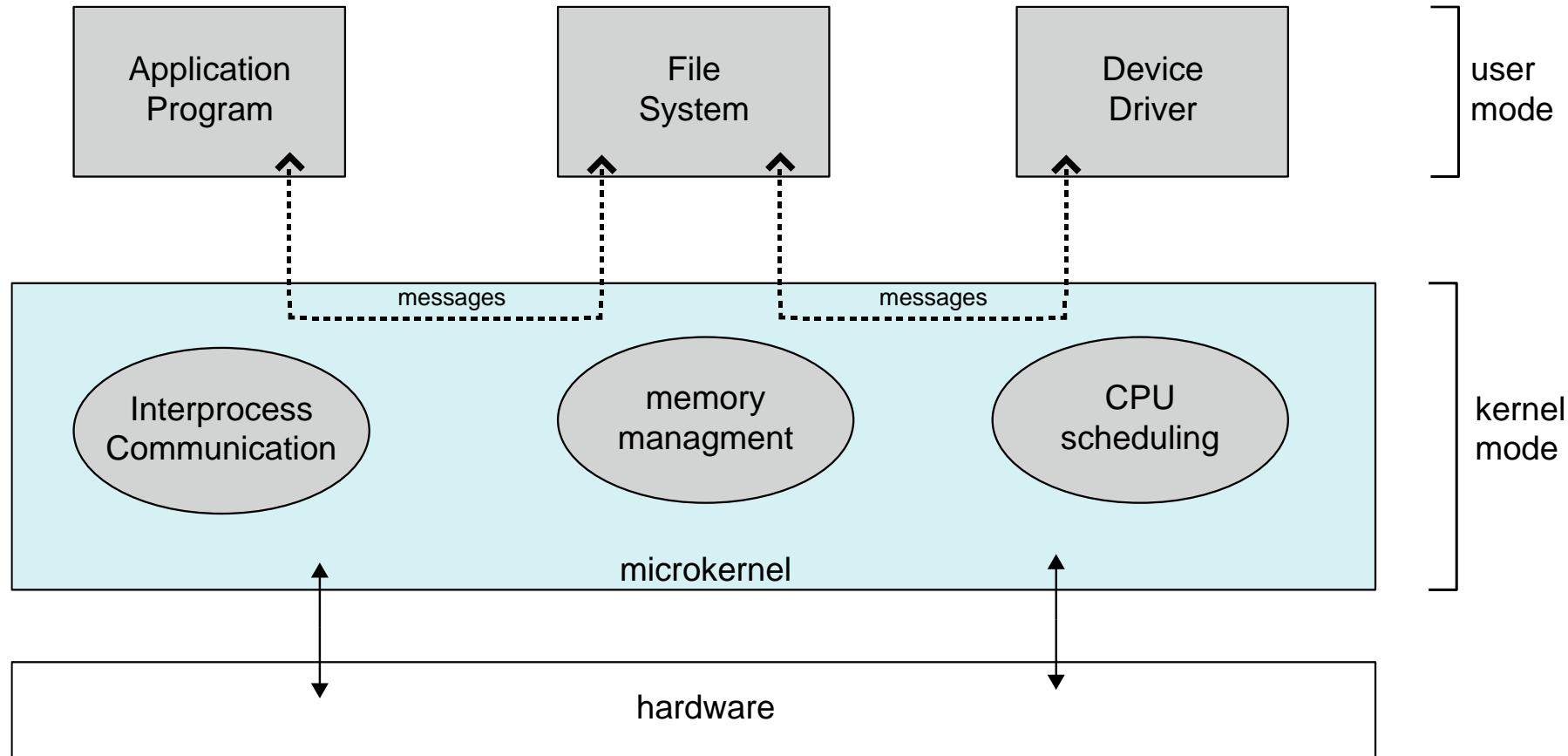
Micro-branduolio (microkernel) sistemos struktūra

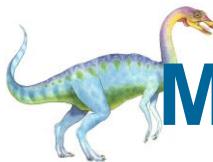
- Perkelia kuo daugiau iš branduolio į vartotojo erdvę.
- **Mach** yra **mikro-branduolio** pavyzdys
 - Mac OS X branduolys (**Darwin**) iš dalies paremtas Mach.
- Vartotojo moduliai komunikuoja tarpusavyje, naudodami žinučių perdavimą (angl. **message passing**).
- Privalumai:
 - Mirko-branduolių lengva išplėsti.
 - Lengviau perkelti OS į kitas architektūras.
 - Patikimiau (mažiau kodo veikia branduolio režimu)
 - Saugiau
- Trūkumai:
 - Spartos sumažėjimas, dėl vartotojo erdvės ir branduolio komunikacijos.





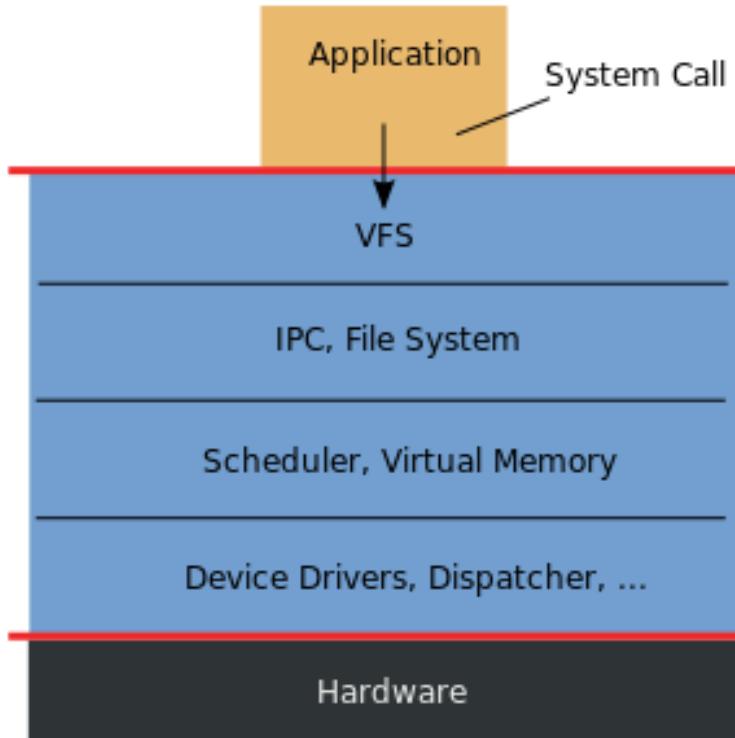
Micro-branduolio sistemos struktūra



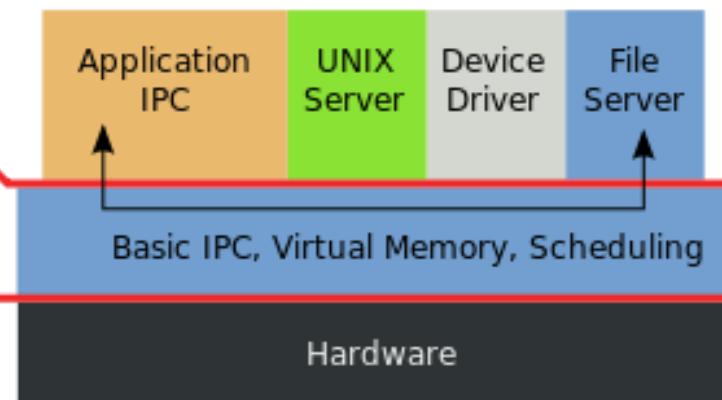


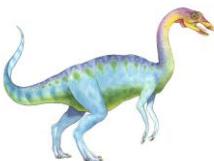
Micro-branduolio sistemos struktūra

Monolithic Kernel
based Operating System



Microkernel
based Operating System

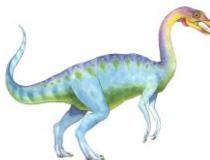




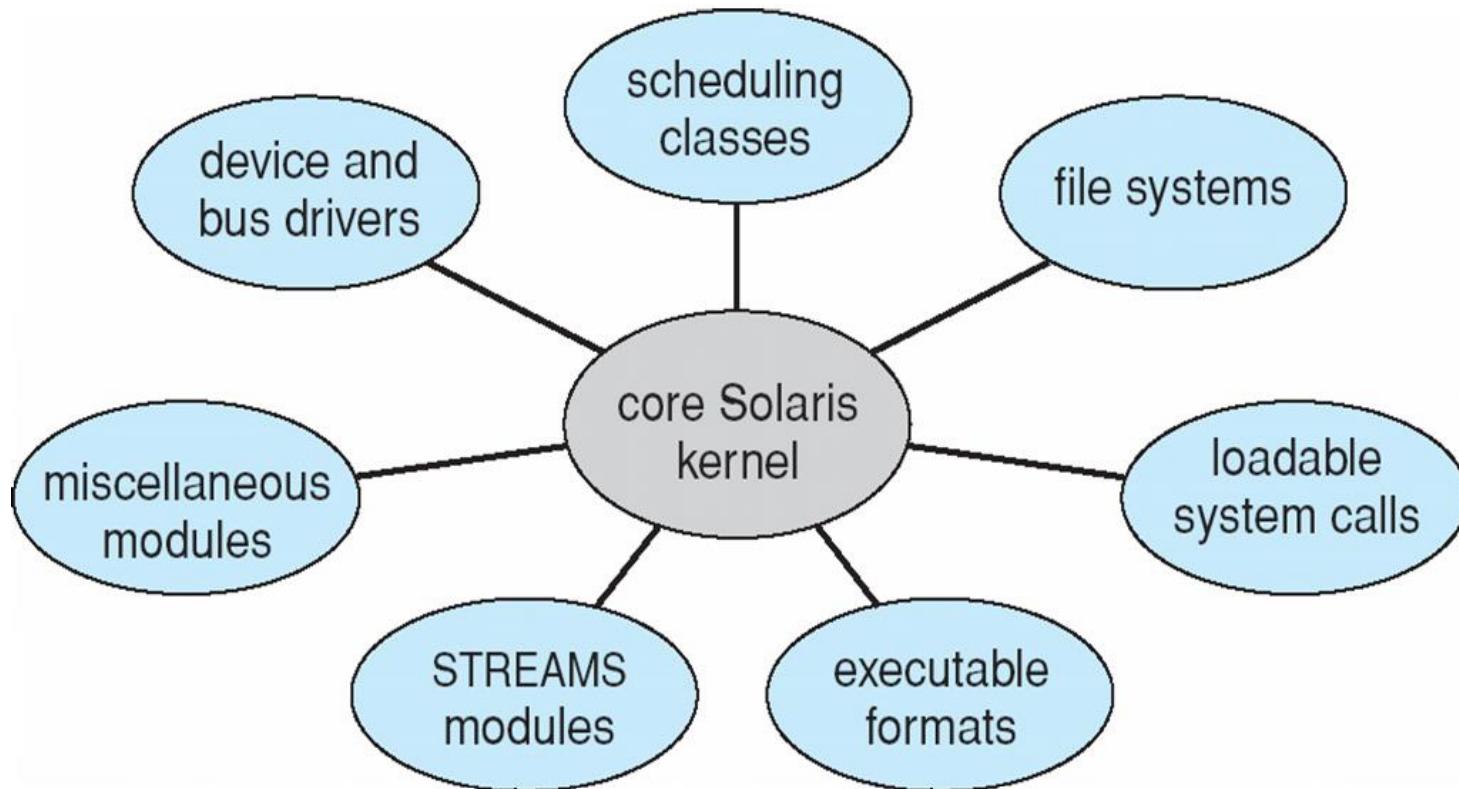
Moduliai

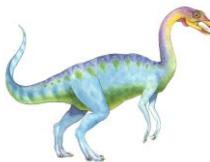
- Dauguma modernių OS įgyvendina paleidžiamus branduolio modulius (angl. **loadable kernel modules**)
 - Naudoja objektinį požiūrį.
 - Kiekvienas pagrindinis komponentas yra atskirtas.
 - Kiekvienas modulis kalba su kitais per žinomas sąsajas.
 - Kiekvienas modulis yra paleidžiamas, kai jo reikia branduoliui.
- Bendrai, panašus į sluoksnius, tačiau daug lankstesnis.
 - Naudojama Linux, Solaris, kt.





Solaris modulinis požiūris





Hibridinės sistemos

- Daugumoje modernių OS nėra taikomas vienas grynas modelis.
 - Hibridinės sistemos apjungia keletą požiūrių, kad išspręsti spartos, saugumo, naudojamumo problemas.
 - Linux ir Solaris branduoliai yra branduolio adresų erdvėje, taigi jos yra monolitinės, plius modulinės dinaminiam funkcionalumo išplėtimui.
 - Windows daugiausia monolitinės, plius micro-branduolio skirtingoms subsistemoms.
- Apple Mac OS X yra hibridinė, sluoksniuota, **Aqua** UI plius **Cocoa** programavimo aplinka.
 - Žemiau branduolio yra branduolys susidedantis iš Mach micro-branduolio ir BSD Unix dalij, plius I/O rinkinys ir dinamiškai paleidžiami moduliai (vadinami **kernel extensions**)





Hibridinės sistemos

- BeOS kernel
 - Haiku kernel
- Syllable
- BSD-based
 - DragonFly BSD (first non-Mach BSD OS to use a hybrid kernel)
 - XNU kernel (core of Darwin, used in OS X and iOS)
- NetWare kernel
- Inferno kernel
- NT kernel (Windows NT 3.1, Windows NT 3.5, Windows NT 4.0, Windows 2000, Windows Server 2003, Windows XP, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, and Windows Phone 8)
- ReactOS kernel
- MikeOS kernel





Mac OS X struktūra

graphical user interface

Aqua

application environments and services

Java

Cocoa

Quicktime

BSD

kernel environment

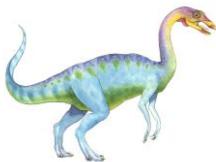
Mach

BSD

I/O kit

kernel extensions





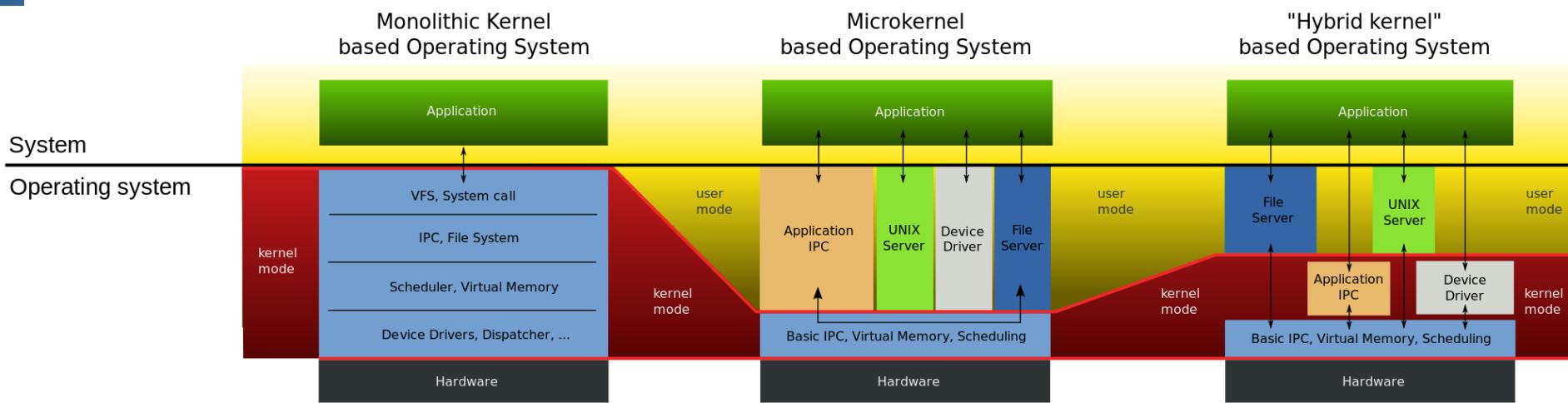
Monolitinis branduolys

- OS architektūra, kurioje visa OS veikia branduolio erdvėje.
- Modelis nuo kitų skiriasi tuo, kad jis aprašo aukšto lygio virtualią sąsają prie kompiuterio techninės įrangos.
- Sisteminių iškvietimų rinkinys įgyvendina visas OS paslaugas, kaip procesų valdymas, vienalaikiškumas, atminties valdymas.
- Įrenginių tvarkyklės gali būti pridedamos prie branduolio, kaip moduliai.





Monolitinis branduolys





Monolitinis branduolys

- Unix kernels
 - BSD
 - ▶ FreeBSD
 - ▶ NetBSD
 - ▶ OpenBSD
 - ▶ MirOS BSD
 - ▶ SunOS
 - UNIX System V
 - ▶ AIX
 - ▶ HP-UX
 - ▶ Solaris
 - OpenSolaris / illumos
- Unix-like kernels
 - Linux
- DOS
 - DR-DOS
 - MS-DOS
 - ▶ Microsoft Windows 9x series (95, 98, 98SE, Me)
 - FreeDOS





iOS

- Apple mobiliųjų įrenginių OS, pristatyta 2007 m., skirta ***iPhone***, ***iPad***, ***Apple TV***, ***iPod***
 - Grindžiama Mac OS X, pridėtas funkcionalumas
 - Nepaleidžia OS X taikomujų programų
 - ▶ Veikia skirtingoje CPU architektūroje (ARM vs. Intel)
 - **Cocoa Touch** Objective-C API programų kūrimui
 - **Media services** sluoksnis grafikai, audio, video
 - **Core services** suteikia debesų kompiuterijos, duomenų bazių savybes
 - Pagrindas grindžiamas Mac OS X branduoliu

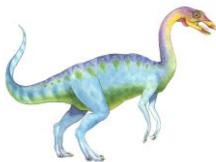
Cocoa Touch

Media Services

Core Services

Core OS





android

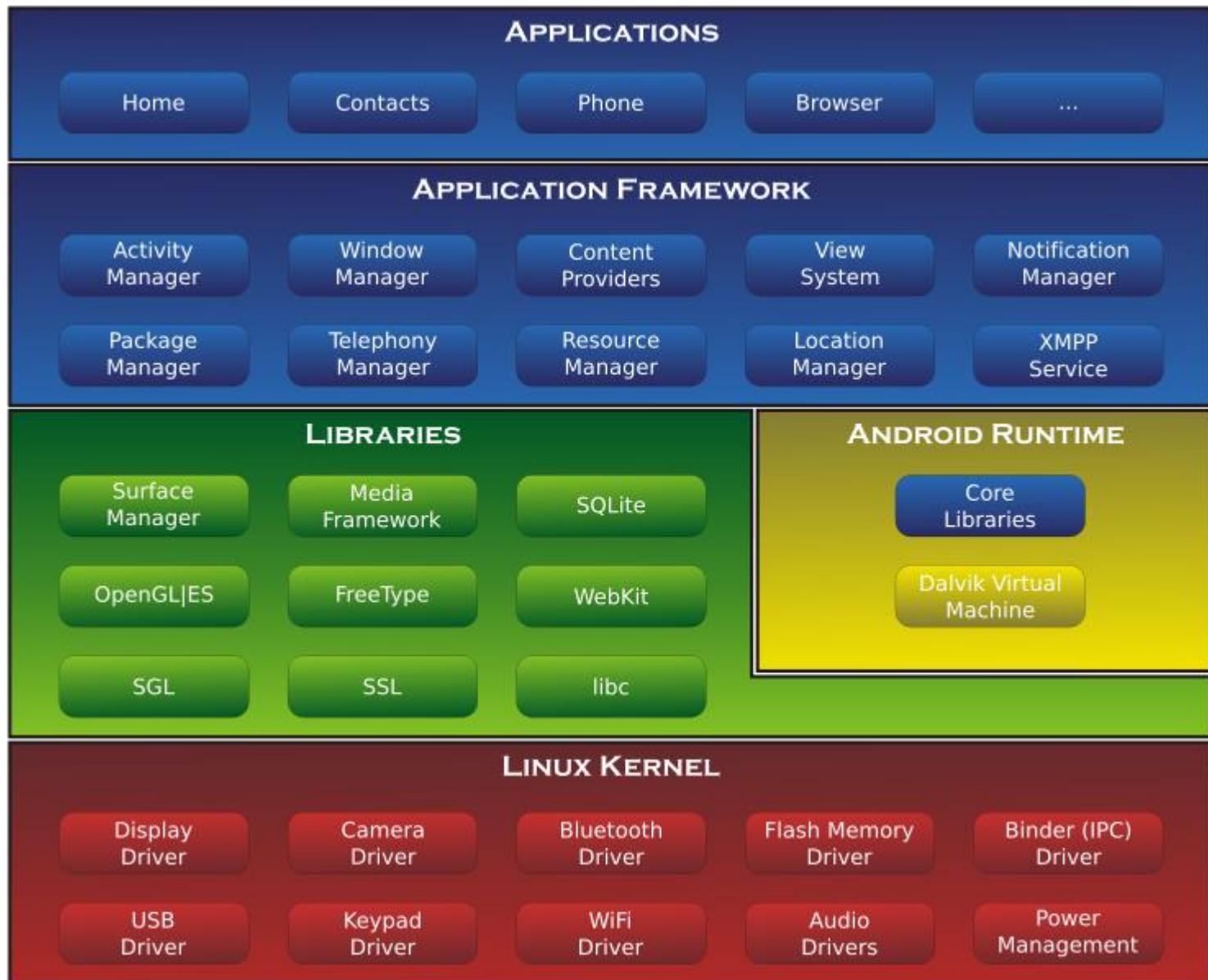


- Vystomas Open Handset Alliance (mostly Google). Pristatyta 2008 m.
 - Atvirojo kodo
- Panašus stekas į iOS.
- Paremta Linux branduoliu, tačiau jis modifikuotas.
 - Suteikia procesų, atminties, tvarkyklių valdymą.
 - Pridėtas energijos valdymas.
- Vykdymo aplinka susideda iš pagrindinio bibliotekų rinkinio ir Dalvik virtualios mašinos.
 - Programos kuriamos Java kalba plius Android API.
 - ▶ Java klasių failai sukompliuojami į Java bytecode, tuomet išverčiami į paleidžiamuosius failus, veikiančius Dalvik VM.
- Bibliotekos apima karkasus interneto naršyklei (webkit), duomenų bazėms (SQLite), multimedia





Android architektūra





Android savybės

- Žinutės
 - SMS ir MMS, Android Cloud To Device Messaging (C2DM),
Android Google Cloud Messaging (GCM).
- Interneto naršyklė
 - Grindžiama atvirojo kodo Blink (anksčiau WebKit) varikliu, su
Chrome V8 JavaScript varikliu. Naršyklė surenka 100/100 Acid3
teste Android 4.0 versijoje.
- Multi-touch
 - Android turėjo pradinį multi-touch palaikymą jau su HTC Hero.
Funkcija buvo išjungta branduolio lygmenyje, siekiant išvengti
Apple patentų lietiminėse technologijose.
- Ryšiai
 - GSM/EDGE, Wi-Fi, Bluetooth, LTE, CDMA, EV-DO, UMTS, NFC,
IDEN, WiMAX.





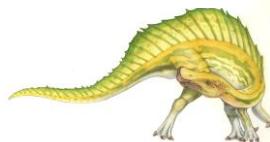
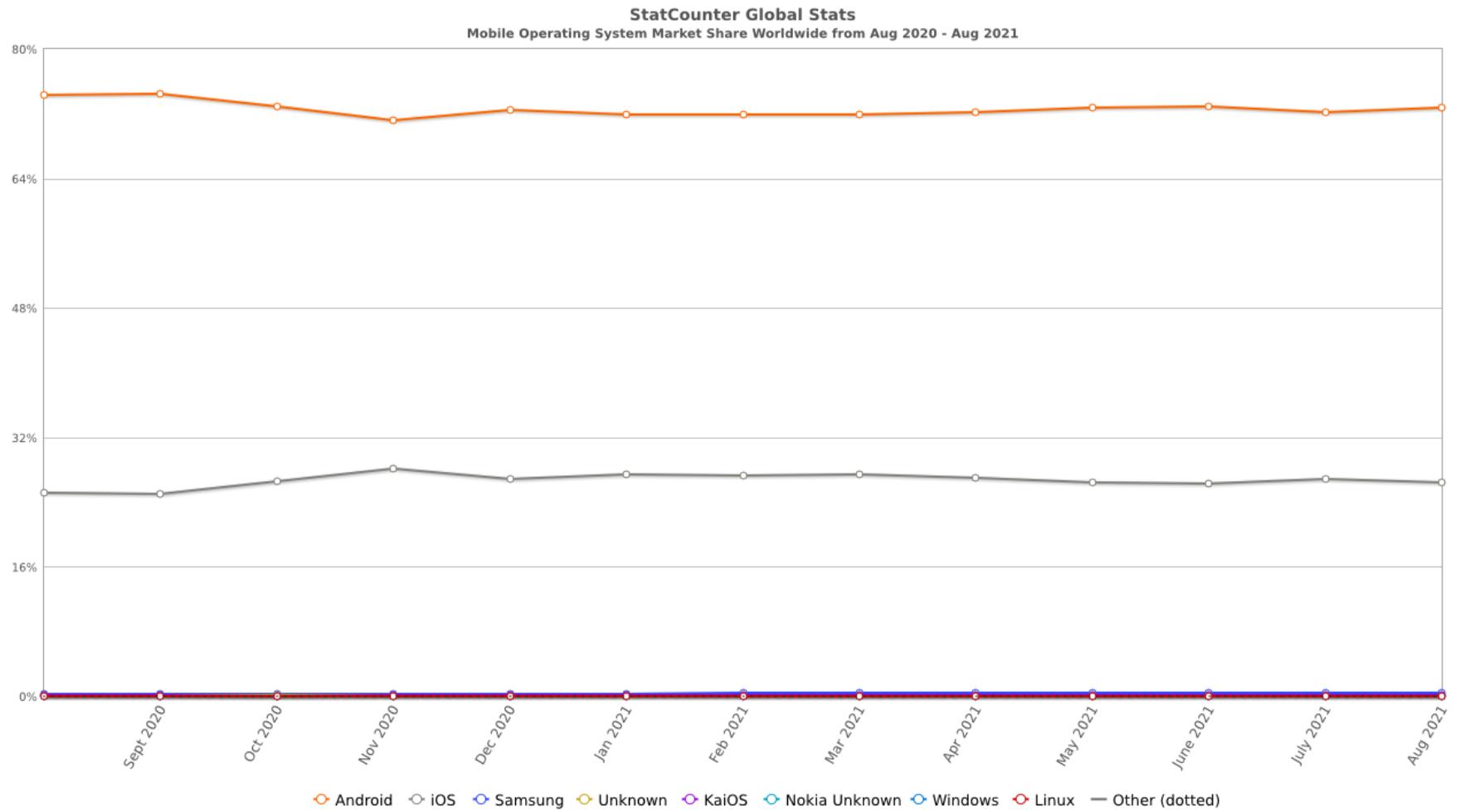
Android savybės

- Transliuojamos medija palaikymas
 - RTP/RTSP streaming (3GPP PSS, ISMA), HTML progressive download (HTML5 <video> tag). Adobe Flash Streaming (RTMP), HTTP Dynamic Streaming, Apple HTTP Live Streaming is supported by RealPlayer for Android.
- Medija formatų palaikymas
 - WebM, H.263, H.264, AAC, HE-AAC (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), MP3, MIDI, Ogg Vorbis, FLAC, WAV, JPEG, PNG, GIF, BMP, WebP.
- Išorinė atmintis
 - Dažniausiai microSD kortelės su FAT32, Ext3 ar Ext4 failų sistemomis. FAT32 yra valdomas Linux Kernel VFAT driver, NTFS, HFS Plus ir exFAT – trečiuju šalių tvarkyklėmis.





Pasaulinė mobiliųjų įrenginių OS rinka 2021





OS Debugging

- **Derinimas (angl. *Debugging*)** yra klaidų paieška ir taisymas.
- OS generuoja ***log files***, turinčius klaidų informaciją.
- Taikomosios programos gedimas gali sugeneruoti ***core dump*** failą įrašant proceso atmintį.
- OS gedimas gali sugeneruoti ***crash dump*** failą turintį branduolio atmintį.
- Spartos derinimas gali optimizuoti sistemos spartą.
 - Kartais naudojami veiklų ***trace listings***, įrašomi analizei.

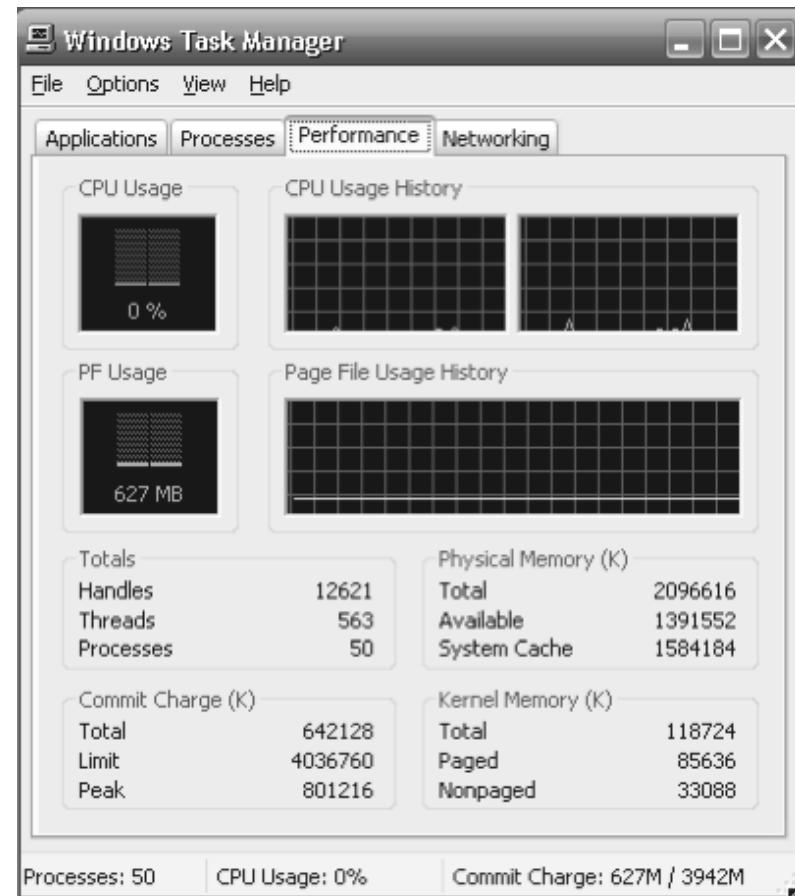
Kernighan taisyklė: “Debugging yra dvigubai sunkesnis, nei kodo rašymas iš pradžių. Taigi, jei kodą rašai taip protingai, kaip įmanoma, pagal apibrėžimą, nesi pakankamai protinges, kad jį debugg’inti.”





Spartos derinimas

- Sparta padidinama pašalinant „butelio kakliuką“.
- OS turi suteikti būdą išmatuoti ir atvaizduoti sistemos veikimo parametrus.
- Pvz., Windows Task Manager





Klausimai

1. Išvardinkite bent tris OS sisteminius servisus, kurie naudingi vartotojui.
2. Išvardinkite bent tris OS funkcijas, kurios išlaiko efektyvų OS darbą.
3. Kokie trys pagrindiniai vartotojo sąsajų būdai?
4. Kas yra sisteminis iškvietimas?
5. Kas yra API?
6. Kaip gali būti perduodami parametrai sisteminiams iškvietimams?
7. Išvardinkite bent 3 pagrindines sisteminį iškvietimų kategorijas.
8. Kuri OS dalis priima sprendimus, kurie susiję su tuo, koks darbas bus atliekamas konkrečiu metu?
9. Išvardinkite žinomas hibridines OS.
10. Kokiuose įrenginiuose veikia iOS sistema?
11. Kokiu metodu komunikuoja mikro-branduoliai tarp procesų?

