Operációs rendszerek

ELTE IK.

Dr. Illés Zoltán

zoltan.illes@elte.hu

Miről beszéltünk korábban…

- Operációs rendszerek kialakulása
- Op. Rendszer fogalmak, struktúrák
- Fájlok, könyvtárak, fájlrendszerek
- Folyamatok
- Klasszikus IPC problémák
- Folyamatok ütemezése
- I/O, holtpont probléma
- Memória kezelés
- Valós idejű rendszerek

Mi következik ma…

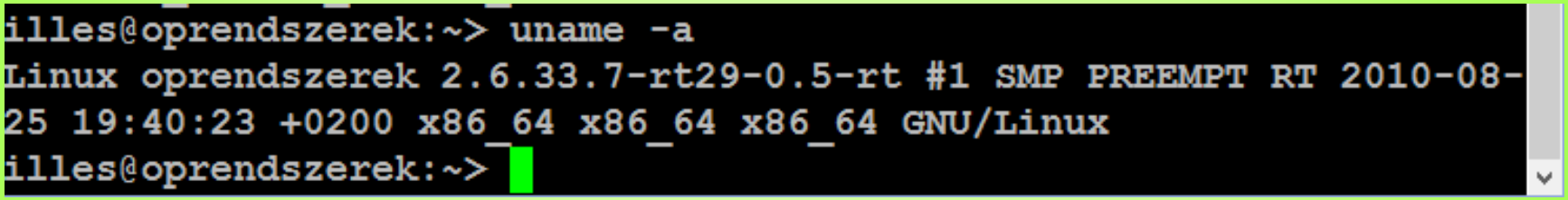
- Modern operációs rendszer környezetek
- Milyen jellemző megoldásokat találunk a mai modern operációs rendszer környezetekben?
  - Linux (Android)
  - Windows

Unix-Linux világ címszavakban

- Bell Labs, Multics utód rendszer, 1970-es évek eleje, UNIX, PDP7,PDP11, Ken Thompson, Brian Kernigham,Dennis Ritchie
- Portable Unix (C nyelv), Berkeley Unix(BSD)
- Két különböző irányzat, BSD, System-V (SVID, AT&T), 1980-as évek vége
- Standardisation, IEEE 1003.1 (ANSI,ISO standard), 1988.
- LINUX- Linus Torwalds, v0.01-1991, v1.0-1994
  - Sok disztribúció(SUSE,Red Hat,Debian,stb)
  - www.distrowatch.com

Linux jellemzők

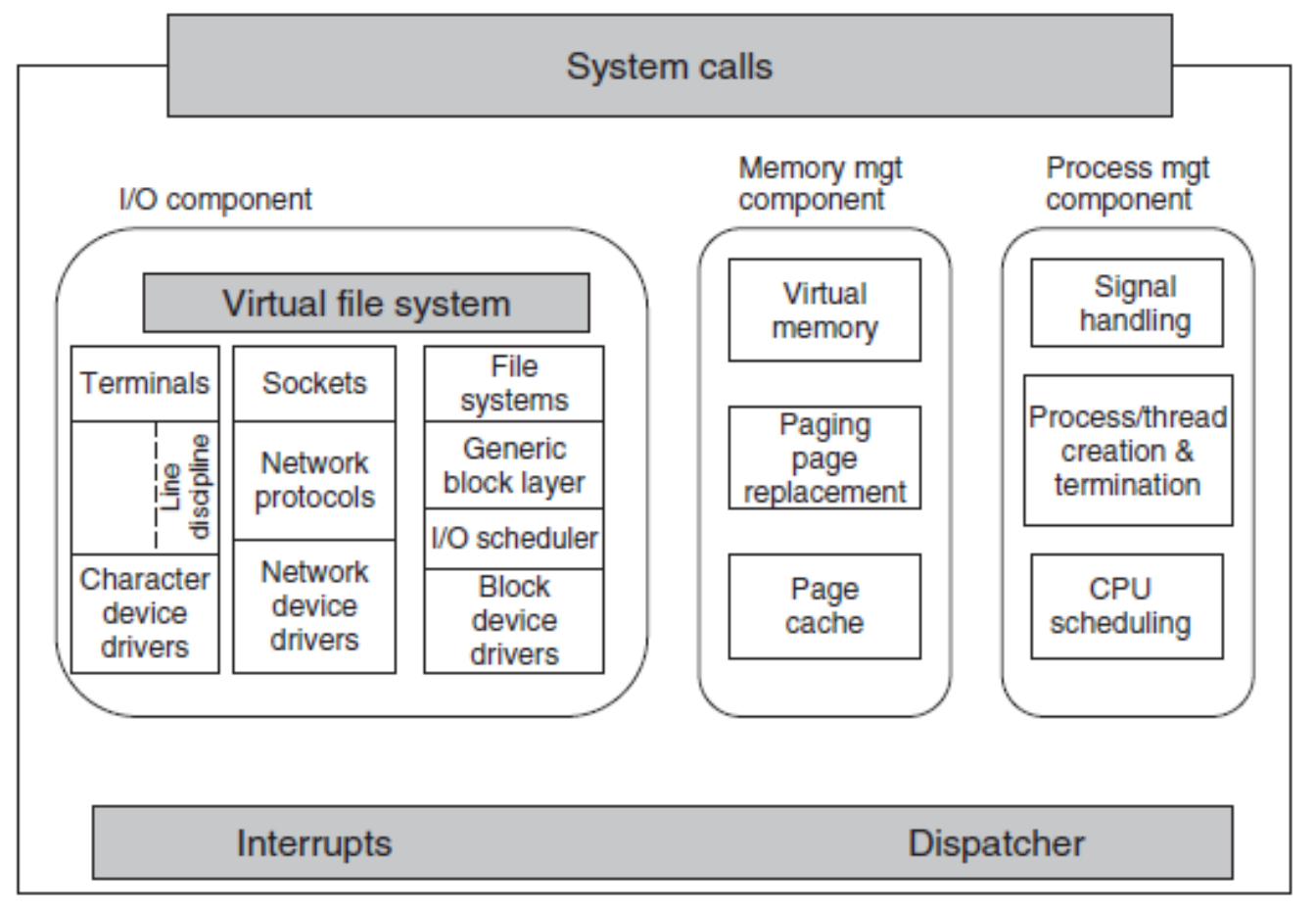
- Hagyományos terminál mód+X-Windowing System (X11)
- Rendszer verzió: uname -a



- Klasszikus réteges felépítés
  - 1.Hardver réteg
  - 2.Kernel (Process, memory,I/O, file system) réteg
  - 3.Standard library (open,close, fprint, fork, exec…)
  - 4.Standard programs (shell, gcc, etc..)

Kernel structure

- User's applications



- Hardware

Folyamatok

- Processes – background processes (daemons)
- Process tree (pstree)– PID, root element is init, PID=1

```
illes@oprendszerek:~> ps -ef|grep init
root         1     0  0 Jan16 ?        00:00:04 init [5]
root      3558     1  0 Jan16 ?        00:00:09 /usr/sbin/sshd -o
PidFile=/var/run/sshd.init.pid
root      3841     1  0 Jan16 ?        00:00:00 /usr/sbin/xinetd -
pidfile /var/run/xinetd.init.pid
illes    13317 12666  0 09:36 pts/0    00:00:00 grep init
illes@oprendszerek:~>
```

- Message passing (pipes, signals,…)

Processes, threads

- A Linux process often calls as task!
  - A task_struct is used to represent any execution context!
  - Main fields: scheduling parameters, registers, signal masks, file descriptor table, memory image, etc.
- POSIX.1defines pthreads kernel threads
  - See: man pthreads
- Clone – system call, creating a thread in the original process, or in a new process
  - The original process can share some of it's data, like address space, file descriptors, signal handling, file system parameters, etc

Ütemezés Linuxban I.

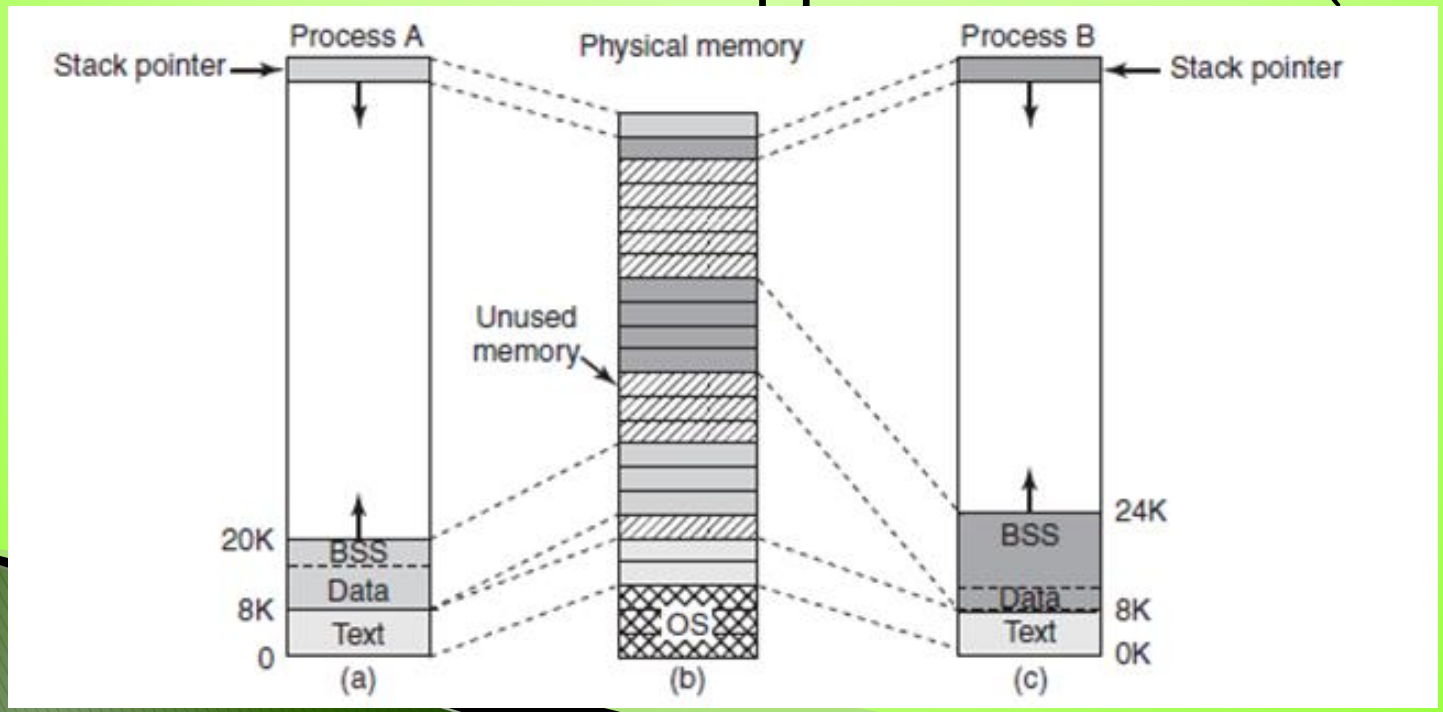
- POSIX1003.4 szabvány (real-time extensions to UNIX) után a 2000-es évekre kétféle ütemezési típus támogatás jelent meg!
  - Preemptív időosztásos, prioritásos ütemezés (klasszikus, 0-39 prioritás osztály, -20-tól +19-ig tartó intervallum, 0 a default, nice utasítás, -20 a legnagyobb prioritás!)
  - Valós idejű ütemezés (0-99 prioritás osztály, 0 a legnagyobb, 99 a legkisebb prioritás, de ezen osztályok mindegyike nagyobb prioritású mint a klasszikus 0-39,)

Ütemezés Linuxban II.

- Statikus (eredeti, alap), dinamikus (folyamatosan változó) prioritás
  - Prioritás módosítás (boost priority)
- Klasszikus időosztásos ütemezés:
  - CFS (Completely Fair Scheduling, Ingó Molnár, Red-Black tree, virtual time slice)
- Real-Time ütemezés
  - Prioritásonként RUNQUEUE!
  - SCHED_RR (Round-Robin)
  - SCHED_FIFO
- Chrt parancs

Memory management

- A process virtual address space contains : code (text), data and stack block
- In the example A and B process shares his code block because A and B same application (e.g:vi), processes can use a mapped file as well!(mmap)



Linux Memory Management

- 32 bit system: 3GB virtual memory size, 1GB kernel data
- 64 bit system: used only 48 bit,(256TB, virtual memory space, 128TB for process, 128TB for kernel data)
- Memory allocation:
  - 1. Buddy algorithm (Dividing in half of memory while the size is uncorrect!) Memory pages: 2^n
  - 2. If smallest one is too large, take a smaller unit and manage it!
  - 3. Use vmalloc!

Page replacement

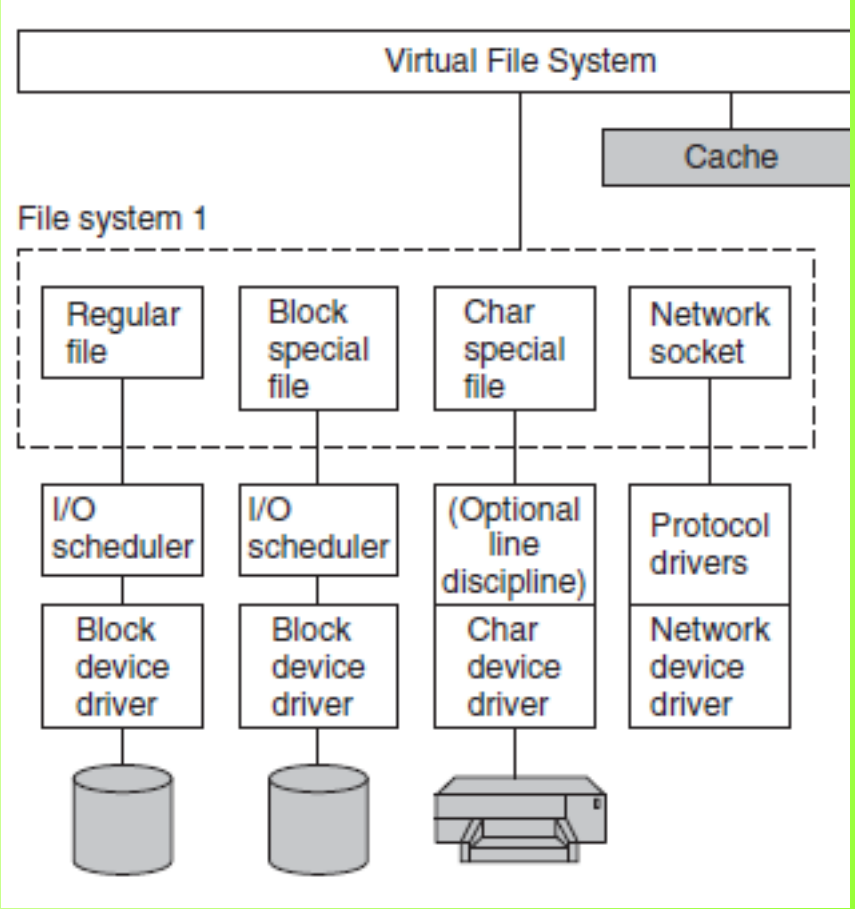
- Page size: 4 kb (in 64 bit system 2MB page system also supported)
- Process ID 2: page daemon, the swapper procces
- Page Frame Reclaiming Algorithm (kswapd)
  - 4 types of pages: unreclaimable(pinned), swappable (must be written back before reclaiming), syncable (must be written back if dirty bit is set), discardable (can be reclaimed immediatelly)
  - PG_active, PG_referenced bits
  - Clock-like algorithm, from the easiest point: first take the discardable, unreferenced pages
  - Active, inactive page list

Input/Output in Linux

- Character special files (terminal connections)
- Block device files (file sytems)
- Networking
  - Berkeley design „socket”
    - If a socket is created we can choose for transmissions a reliable byte stream protocoll (TCP)
    - Or for an unreliable packet-oriented transmission we can choose UDP!
  - Once a connection is ready between source and destination processes, it functions analog to a pipe!

Linux I/O architecture

- To reduce repetitive disk-head movements, there is used block I/O scheduler
- It is similar as „SSTF” schedules (File systems)
  - It can lead starvation!
  - 2 lists,
    - 1 is ordered by address of sector of disk request
    - 2 is the deadline list (anti starvation)



Linux file systems

- Tipycal Unix style file system based on index tables (i-node tables)
- Shared lock, exclusive lock for an entire file, or for a part of file!
- Virtual File System(VFS) has 4 object
  - Superblock, dentry(directory), i-node, file(in memory file representation)
- Most popular Linux file system: Ext2FS
- Dentry cache- for quick search
- File descriptor table does not directly map to a file(i-node)
  - A particular file descriptor corresponds to an open file desciptor table element-contains the file position, R/W mode,i-node

Ext3FS, Ext4FS

- However the most general FS is Ext2FS, tipycally that successor (Ext3FS) is used!
  - Check your FS: df -T

```
Filesystem     Type   1K-blocks      Used Available Use% Mounted on
/dev/sda2      ext3   101139356  22144884  73856888  24% /
devtmpfs   devtmpfs     8028468        92   8028376   1% /dev
tmpfs          tmpfs    8028468        88   8028380   1% /dev/shm
```

- Ext3FS is a journaling file system.
  - Basic idea: maintain a „journal”(log), which describes all operations in sequential order!
  - Benefit: At any unexpected case the system recognise it and apply the file system changes based on journal log!
- Ext4FS – supporting extents, gives faster support at larger file and file system size!

Security in Linux

- Same as in a general UNIX
  - Base security is a 3x3 rwx control!
  - Additional: SETUID,SETGID and STICKY BIT.
- Particural rights: setfacl, getfacl
- User ID stored in /etc/passwd file
  - A user's password stored in /etc/shadow file in hashed format!
  - Manually to check a password:
    - 1. Enter the pw as a text.
    - 2. Get the user's shadow enry.
    - 3. Getting th „salt"
    - 4. Crypt the pw.
    - 5. Compare it with the existing shadow one!

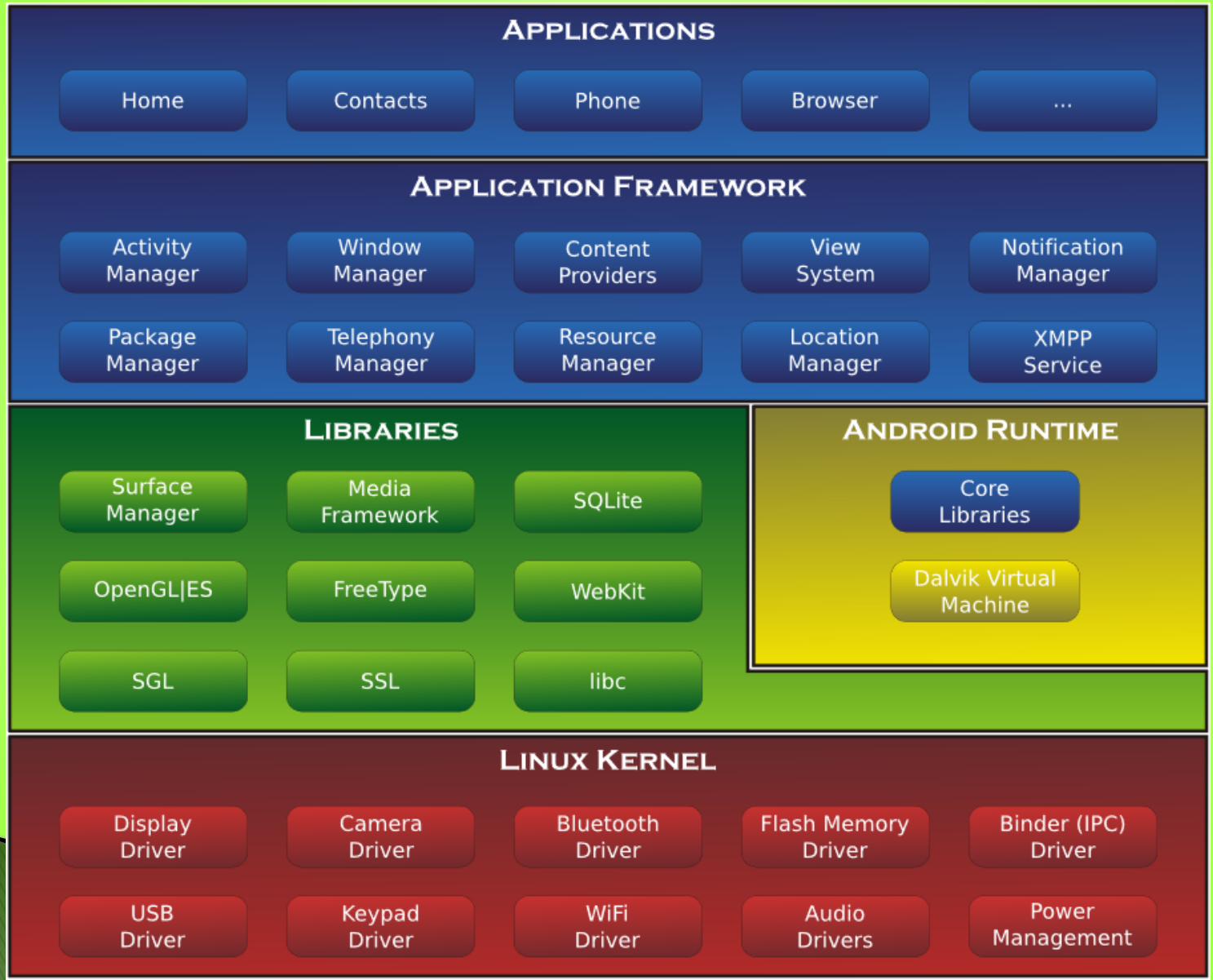
Android

- New operating system, designed to run on mobile devices.
- Kernel and most of low level libraries written in C, C++.
- Generally all other part is developed in Dalvik Java. (Quicker VM, more efficient memory usage)
- In 2005 Google acquired Android Inc.
- Today the most popular mobile (phone, tablet, TV, etc.) system.

Android extensions

- Wake lock – manage how the system goes to sleep
- Out-of-Memory Killer – replacesthe traditional Linux one, it is more aggressive
- Binder IPC- transaction based RPC
- Android application- package, .pkg, contains not only code, but resources, manifest, etc.
- Application activity – application part which interacts with user via UI(event-handler)
- Application Sandboxes – a new UID for an app according security reason.
- Process modell – every process is created and managed by Dalvik's zygote modul

General Android architecture



Android today

- Ver.3 is for tablets!

Verzió	Elnevezés	Kiadás dátuma	API-szint	Eloszlás[14]
6.0	*Marshmallow*	2015. október 5.	23	0.3%
5.1-5.1.1	*Lollipop*	2015. március 9.	22	10.1%
5.0.0-5.0.2		2014. november 3.	21	15.5%
4.4	*KitKat*	2013. október 31.	19	37.8%
4.3	*Jelly Bean*	2013. július 24.	18	4.1%
4.2.x		2012. november 13.	17	13.9%
4.1.x		2012. július 9.	16	11.0%
4.0.3–4.0.4	*Ice Cream Sandwich*	2011. december 16.	15	3.3%
2.3.3–2.3.7	*Gingerbread*	2011. február 9.	10	3.8%
2.2	*Froyo*	2010. május 20.	8	0.2%

Android's future

- However Android is a general full porpuse operating system, but it is running mostly mobile device.
- Android – general computer , maybe soon.
- …
- On other hand, there are other Linux successor mobil distribution too: e.g: Mobile Ubuntu
- …

Windows világ

- MS-DOS 1.0 – 1981
- Windows 3.0,3.1 – 1990,1992
- Windows 95, NT – 1995
  - Elválik az MS-DOS alapú (kliens) és NT alapú (szerver) vonal!
- MS-DOS alapú (kliens) rendszerek:
  - Windows 98, Windows ME -1998, 2000
- NT alapú (kliens, szerver) rendszerek:
  - Windows 2000 (kliens, szerver)
  - Windows XP (kliens)-2002, Windows 2003(szerver)
  - Windows Vista, 7(2006,2009), Windows szerver 2008
  - Windows 8,8.1, 10(2012-2015),Win2012 szerver

Windows programming elements

- MinWin approach, from win8.1, win10
  - Same the most of the core, binaries for all version including WP!
- Win8.1 removes POSIX support!
- New WinRT (Runtime) API (replaces Win32)
- NT namespace holds OS names, objects (e.g. device objects) created during boot, stored in kernel's virtual memory.
  - This part is marked as permanent!
- API fearures
  - Unicode, WoW(Windows on Windows) functions,ACL, journaled NTFS,I/O subsystem manage, GUI fnctions, etc.

Windows registry

- During boot the NT namespace is created.
  - How it is created? Where is the configuration parameters stored?
- The registry holds these information!
  - Registry files (hives) stored in directory Windows\system32\config
    - System – HKLM\System
    - SAM – HKLM\SAM (Security Access Manager)
    - Etc.
  - In earlier Windows versions this parameters was stored in .ini files, same as UNIC config files!
  - To explore registry we can use API calls or regedit.exe, or PowerShell!
  - Over time of Windows versions the size, the disorganization was evolved, so be carefull to make any modification in it!

System structure

- 1. Hardware layer (CPU, memory, devices, BIOS, etc)
- 2. Hypervisor layer (if exist, not necesserry), every OS runs in its virtual machine.
- 3. HAL - Hardware Abstraction Layer, holds abstracts low level hw detailes, registers, timers,DMA,etc.(hal.dll)
- 4. NTOS Executive layer(contains I/O manager, process manager, memory manager, notification, etc)
- 5. NTOS kernel layer (Deferred Procedure calls, ISR, APC)
  - ntoskrnl.exe

Processes in Windows

- Processes can be groped – called job! (batch processing feature)
- Each process has a user mode data: PEB (Process Environment Block)
  - It includes loaded modules, environment data, etc.
- Every process starts with one thread! Process acts as a „thread container”!
  - Later new threads can be created dinamically as needed!
  - TEB (Thread Environment Block) – user data for thread

Interprocess communication, syncronization

- IPC functions:
  - Pipe, named pipes
    - Byte and message type pipes!
  - Sockets
    - Mailslots (for OS/2 compatibility)
  - Shared files (memory)
  - RPC
  - There are no significant differences to Unix.
- Syncronization
  - Semaphores, mutexes, critical regions, notification events, syncronization events.

Scheduling in Windows

- It uses a priority based scheduling avoiding starvation! (Higher priorities first!) (Dynamic Fair-Share Scheduling)
- There are 32 priorities in Windows!
  - 0 – Zero page thread
  - 1-15 User priorities
  - 16-31System priorities (Real time class)
  - A thread has a base priority and a current priority! (current >=base)
  - Windows maintains 32 lists of threads!
  - Avoiding starvation and for other reasons the kernel boosts the base priority of a thread! (Actual priority always <16)
  - Priority inversion – A lower and higher thread priority will be changed avoiding unnecessary semaphore waits!

Memory management

- In 32 bit environment each virtual address space is 4GB. (In 64 bit longer, depending OS version)
  - Tipycally 2GB User space+2GB shared OS system calls,etc.
  - Pagefile.sys, on system volume.
  - Normally 4kb page size (it can be max. 2MB)
- Memory manager focuses to processes!
  - For a process MM creates a Virtual Address Descriptor(VAD) entry, with 4 data (range of mapped address, backing store region, backing store map, permission)
  - VAD is organised as a ballanced tree (like B-R tree)

Paging – Page faults

- On demand paging based on page faults!
  - A page table entry is a 64 bit long field!
    - Global page, Dirty bit,present bit, Accessed bit,etc
- 5 categories of page faults
  - The referenced page is not comitted (invalid operation)
  - Access to a page is restricted by permission.
  - A shared copy-on-write page was modified.
  - The stack needs to grow.
  - The referenced page is committed and not mapped in! (This is the normal page fault!)

Page replacement algorithm

- Based on working set concept!
  - Each process needs a min-max page number for ideal work.
- 3 working set manager activity:
  - Lots od memory available
  - Memory getting low
  - Memory low (reducing working sets)
  - Working set manager runs appr. every second.
- 4 physical memory reference list:
  - Free, Modified, Standby, Zeroed

I/O operations

- Supporting automatic device discovery.
  - Some devices no needs specific driver!
- DDK – for creating new device driver
  - Driver verifier
  - Windows Driver Foundation (WDF)
  - UMDF – User-mode driver framework
  - KMDF – Kernel mode driver framework
- See books about WDF for more detail!

NT File System

- Supports 255 long file names, 32767 long full path!
- Each NTFS volume a set of blocks (def block size 4kb)
- Index table based structure- main tableis MFT(Master File Table)
  - MFT contains 16 records.
- Journaled file system, supports encryption,compression, soft RAID functionality.

Köszönöm a figyelmet!

zoltan.illes@elte.hu