OS prednaska c2

**HISTORIA OS**

**1.generacia (1940-1955)**

- Plugboard – pouzivaju sa mechanicke rele, vakuove trubicke (elektronky) a programovacie dosky

- neda sa hovorit o existencii OS v pravom slova zmysle

- napr Enigma

- vtedy to bolo tak ze prakticky kazdy pocitac je postaveny na mieru a vyzaduje velmmi drahe skolenie personal atd

**2.generacia (1955-1965)**

**- transistor (1950)** – vypoctove systemy su spolahlivejsie. Siria sa aj do korporacii

-mainframe

-programator naprogramoval program na dierne stitky, ktore sa potom precitali na magneticku pasku, ta paska sa preniesla do vacsieho pocitaca a spracovala sa – input tape, system tape, output tape

System tape – da sa povedat ze prvy OS, mal za ulohu nacitat prvy program a spracovat ho

-vtedy este neexistovali nejake vyssie programovacie jazky, maximalne assembler a fortran

**3.generacia (1965-1980)**

- integrated circuits – tranzistory sa zmensuju a zdruzuju do integrovanych obvodov

- multiprogramming – pocitac udrziava v pamati viac uloh, niekolko slotov pamate rovnakej velkosti. Pripravime si viacero uloh do zalohy, ak by dana uloha ktora sap rave vykonava cakala, nech sa moze vykonavat ina

- spooling – pocitac zaraduje nove ulohy do hlavnej pamate akonahle sa nejaky slot uvolni

- batch systems – dovtedajsie systemy spracovali ulohy davkovo. Odovzdanie spracovanie a vyhodnotenie ulohy moze trvat niekolko hodin

- mainframe mal pamat na viacero uloh a operacny system

- timesharing – variant multiprogrammingu, kde sa striedaju pouzivatelia namiesto uloh

-sucasne moze pracovat a zdielat system mnoho pouzivatelov

-obrat vykonania ulohy sa skracuje z hodin na minuty

-terminals – terminal je sposob ako pouzivatel zadava prikazy mainframe-u. implementacia timesharing-u

-minicomputers – “mensi” pocitac urceny na spracovanie jednoduchsich uloh

-OS/360 neskor MULTICS->UNIX->chaos->POSIX->MINIX->Linux

-UNIX bol velmi popularny, kazda univerzita si chcela spravit vlastnu verziu

**4.generacia (1980- )**

-Large Scale Integration (LSI) – jednotlive cipy obsahuju tisicky tranzistorov

-Personal Computer (PC)

-vznikol Windows, Linux, MacOS

-Xerox vymysleli okna, graficke rozhranie ale nevedeli co s tym tak Steve jobs to vyuzil

**5.generacia (1990- )**

-prvy smartphone, ktory kombinuje telefon a PDA uviedla Nokia, prvy OS v tychto systemoch bol Symbian

^bolo to len tak pre zaujimavost, nedolezite

**Klasifikacia OS**

* Mainframe OS
* Server OS (Multiprocessor OS)
* PC OS
* Embedded OS (Real-time OS)

**Mainframe OS**

* Su urobene tak aby spracovavali velky pocet jednoduchych uloh, Job-ov
* Su vyladene a stabilne systemy

**Server OS**

* Su nastavene tak aby spracovavali velky pocet uloh

**Multiprocessor OS**

* Superpocitace, velky pocet procesorov zdruzenych do Clusterov

**Embedded a Real-time OS**

* Skor na nejaku monotonnu cinnost, napriklad v chladnickach, alebo na meranie teploty v elektrarnach

**Klasifikacia podla struktury**

* Monoliticke – napr. Windows, Linux atd
* Microkernel – jadro OS je co najmensie

**Strucny opis HW**

(CPU->Memory(RAM))->Video controller->keyboard controller->USB controller->Hard disk controller

**CPU**

* Mozog vypoctoveho system
* Dostane instrukcie a vykonava to co ma, nasledne da vystup
* Rozhranim pre OS su:
  + Mnozina instrukcii
  + Register
  + Rezimy CPU
* Situaciu komplikuje architektura CPU
  + Pipeline
  + Multithreading
  + Multicore
  + Prerusovaci podsystem

Sorry nedaval som pozor som kokot

**Stavebné kamene OS**

* Procesy
* Adresny priestor
* Suborove systemy
* Pouzivatelske rozhrania
* Systemove volania
* Manazment I/O (input/output)

**Adresny priestor**

* Je vsetka potencialna pamat adresovatelna adresno.datovou zbernicou procesora
* Fyzicky adresny priestor – je vsetka realne existujuca pamat adresovatelna adresno-datovou zbernicou procesora
* Adresny priestor je vacsinou daleko vacsi ako mame realne k dispozicii
* Jednotlive procesy musia byt navzajom izolovane
  + Existuje ochranny HW, ktory brani zasahu do pamate procesu
  + Tento HW riadi OS
* Process teoreticky moze vyuzit cely adresny priestor
* **Virtualna pamat –** je concept OS zabezpecujuci pristup k celemu adresnemu priestoru v limitovanom fyzickom adresnom priestore
* Bezny sucasny processor moze mat adresny priestor s velkostou do 2^64 B (2^30 \* 16 GB)
* Bezne PC ma RAM o velkosti do 2^34 B (16 GB)

**Suborove systemy**

* **File (Subor)** – je abstrakcia Os, ktora umoznuje pracu s datami pouzivatela bez nutnosti blizsie poznat zariadenie na ktorom su data ulozene. Subor je vlastne struktura ktora obsahuje informacie o blokoch disku, ktore mu patria. Subor moze byt vytvoreny, zmazany, otvoreny, zavrety, citany a zapisovany. Vsetky operacie nad suborom implementuje OS prostrednictvom systemoveho volania
* **Directory (adresar)** – je vo vseobecnosti specialny subor ktory obsahuje informaciu o vsetkych suboroch a inych adresaroch, ktore su pod nim vedene
* **File System (suborovy system)** – je struktura adresarov s ulozenymi subormi. Z pohladu OS je File System abstrakciou so vseobecne znamym API. (Application Programming Interface|Rozhranie pre programovanie aplikacii)

**Shell a GUI**

* **Shell** – je program urceny spristupnujuci funkcie OS. V zasade sa jedna o prikazovy riadok, kde pouzivatel zadava prikazy v textovej podobe a dostava odpovede rovnako ako text. Shell nemozno brat ako sucast OS ale bez neho by sme si nepomohli. Shell definuje jednoduchy sposob ako spustat dalsie procesy a presmerovavat data medzi procesmi
* **GUI** – graficke rozhranie

**Systemove volania**

* **System Call (systemove volanie)** – je specialna funkcia, ktora je sucastou abstraktneho rozhrania (API) OS. Toto API bezne najdeme v tzv. Knizniciach OS. OS a HW vieme ovladat len prostrednictvom tohto API. Centrom systemoveho volania je vykonanie specialnej instrukcie TRAP, ktora zabezpeci prepnutie do privilegovaneho rezimu procesora. Po prepnuti do tohto rezimu OS vyhodnoti ako systemove volanie sa zavolalo a nasledne ho vykona
* **POSIX** – je standard ktory presne vymedzuje minimalne rozhranie, ktore by mal kazdy OS podporovat. Standard urcuje sadu systemovych volani a ich formu

DU do dalsej prednasky

* Zoznamujte sa s prikazovym riadkom!!!!
* Precitat kapitoly 2.1, 2.2 a 2.4 z Tanenbauma. Pozrieme sa na procesy