

1) Operační systém

a) Definujte pojem operační systém

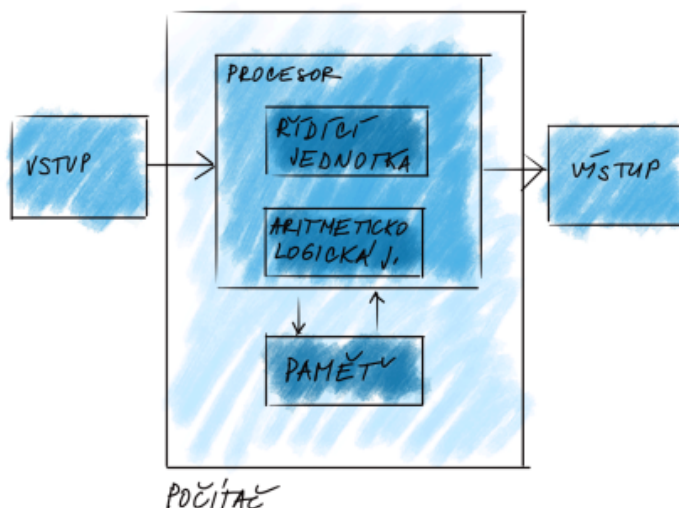
- Operační systém je software, který spravuje počítač (hardware), popřípadě umožňuje uživateli pracovat s počítačem.

b) Uveďte vlastnosti a funkce

- Abstrakce hardware (zapouzdření): při alokaci paměti (malloc) nás nezajímá typ úložiště etc.
- Řízení přístupu k operační paměti: každý program si hraje na svém písečku, nesmíme dopustit aby si mezi sebou bez dovolení sahali na data (jsou tajné), navíc by mohlo dojít k chybě
- Řízení přístupu k procesoru: moderní CPU zvládnou několik programů naráz, nepracují všechny najednou - záleží na schopnosti CPU, ostatní procesy musí čekat
- Řízení vstupů a výstupů: klávesnice, myš, tiskárna, monitor
- Běžové prostředí pro programy: poskytování API, knihoven, instalace nových programů bez restartu, procesy na sebe navazují, komunikace mezi aplikacemi
- Uživatelské rozhraní: některým uživatelům nevyhovuje command line, amateři lol
- Rozhraní pro autory aplikací: API od systému pro vykreslování UI, knihovny etc.
- Zabezpečení systému a správa identit: několik uživatelů na jednom PC, jejich práva
- Detekce chyb, záznam událostí a obnovení z chyb: OS může detekovat vlastní chyby (proces na pozadí co na ně čeká), aplikace by měla v krajním případě spadnout a neměl by se sesypat celý operační systém
- Automatická aktualizace software, napojení na distribuční centra SW: Windows update, sudo apt update, sudo apt upgrade (balíkový systém)
- Návod, podpora uživatelů i vývojářů: nápověda (nikdy jsem nepoužil), internetové fóra, dokumentace
- Řízení výkonu systému: práci chceme rovnoměrně rozdělovat mezi procesory (jádra, I guess), aby všechno pracovalo na 100%

2) Hardware

a) Základní schéma počítače a vnitřní zapojení součástí



b) Co je procesor, jak souvisí architektura procesoru s operačním systémem

- Vlastnost: počet bitů, které používá pro určení adresy v paměti RAM (v dnešní době 64 bitové)
- Výpočty probíhají v ALU (aritmeticko-logická jednotka)
- Součást ALU jsou datové registry procesoru (vnitřní paměťové buňky procesoru, nejrychlejší paměť v PC, přístupová doba odpovídá frekvenci procesoru)
- Řídící jednotka synchronizuje chod ALU s obsluhou paměťové sběrnice, vstupy a výstupy
- Řídící jednotka připravuje instrukce a příprava ALU pro vykonání instrukce
- Obě jednotky jsou složeny z logických obvodů
- Procesor je vybaven periferiemi = přístup ke vnějším zařízením

c) Co je přerušení a jak je využívá operační systém

- Procesor přeruší svou původní činnost a věnuje se zpracování přicházející události (zdrojem událostí jsou například periferie procesoru připojené na jeho vstupy a výstupy)
- O "přerušení" žádají periferie a stará se o ně v procesoru "řadič přerušení", který je řadí do fronty podle důležitosti
- Existují i softwarová přerušení, obslužný kód je součástí operačního systému
- Procesory x86 mají instrukci INT (s parametrem rozsahu 0-255)
- Procesory ARM mají instrukci SWI (software interrupt)

d) Chráněný a nechráněný režim procesoru

- Někdy se též říká "režim jádra/kernel" a "uživatelský režim"
- Nechráněný režim: libovolně bez omezení lze měnit hodnoty stavových registrů a používat všechny instrukce z instrukční sady, lze přistoupit k libovolné datové buňce v operační paměti
- Chráněný režim: omezený přístup k RAM, část je skrytá a nemá R/W práva
- Tento koncept z OS dělá big bossa dat
- Nové Intel procesory mají dokonce 3 režimy, ARM 8 režimů
- Obsluha přerušení vždy probíhá v nechráněném režimu

3) Architektura OS

a) Jádru OS, typy jádru a jejich vlastnosti

- Běží v nechráněném režimu (přístup ke všemu)
- Startuje jako jedna prvních komponent OS
- Monolitické jádro = dělá všechno samo, ovšem nesmí spadnout
 - Unix, Linux
 - Rychlé, velké (stovky MB, pokud specifikujeme zařízení tak desítky MB)
 - Pracuje v souvislém paměťovém prostoru
 - Plug&Play, změny za chodu se řeší pomocí modulů (mohou být spuštěny bez vypnutí jádra) => probíhá fragmentace paměťového prostoru
- Mikrojádro = všechna činnost je realizována jinými programy
 - Jádro má jen to nejnnutnější a ostatní věci jsou odsunuty do samostatných programů, které s OS spolupracují
 - Mikrojádro v sobě má jen: plánovač procesů (scheduler), správa paměti, modul pro zajištění komunikace mezi procesy (interprocess communication "IPC")
 - Mohou mít problém s výkonem, ovšem není prokázáno studií
 - Používal to Blackberry, teď to má Ford v automobilech
- Hybridní jádro = směs předchozích dvou
 - Windows obsahuje od verze Windows NT
 - Většina služeb pracuje v nechráněném režimu
 - Některé ovšem běží v chráněném: např. Session Manager (relace uživatelů), Authority Subsystem Service (ověřování uživatelů) a další systémy emulující další běhová prostředí (POSIX, Linux Subsystem for Windows (Ubuntu 14.0.4 LTS))

b) Volání systému

- Program nemůže pracovat sám se soubory (neví adresu, práva etc.), zavolá proto OS aby mu s tím pomohl přes API s nějakými parametry

c) Propojení API systému s kódem aplikací

- API přidáme do kódu přes nějakou knihovnu, třeba jako v C " #include "
- Tato API by měla být zdokumentovaná a knihovna s ní komunikuje

d) Využití chráněného a nechráněného režimu pro práci jádra OS

- Chráněný režim: Aplikace zavolá API s parametry
- Nechráněný režim: Jádro si ověří zavolání API, oprávnění, pokud je vše správně tak následuje další krok nebo chyba
- Nechráněný režim: Vykoná se požadavek API a vrátí se výsledek aplikaci
- Chráněný režim: Aplikace pokračuje ve své práci, nyní třeba se souborem něco něčím co dostal od jádra
- PS: (Celé jádro pracuje v nechráněném režimu)

e) Příklady konkrétních architektur (Android, Windows, Linux)

- Jádro se musí přeložit pro specifický OS, v makrech (parametrech) se nastavují požadované funkce Linux po přeložení
- Linux: Monolitické, může se k němu připojit modul
- Windows: closed-source,
- Zephyr: open-source, funguje podobně jako Linux, jádro systému a služby se překládají spolu s kódem aplikace (do stromu zdrojových kódů se přidá složka app obsahující kód aplikace, výsledkem překladače je jeden soubor)

4) Procesy

a) Co je proces?

- Proces je konkrétní spuštěný program, jenž má přidělen prostor v operační paměti
- Má v RAM kód a všechna svá data
- Jeden program dokáže spustit více procesů
- Jsou to vlastně jen data v paměti RAM
- Procesy mají PID (process identifier) + PCB (dodatečné informace o procesu)

b) Životní cyklus procesu

- 1) Spuštění/Starting: požádání o spuštění procesu, kopírují se data z úložiště do paměti
- 2) Připraven / Ready: čekání na procesor
- 3) Běžící / Running: byl přiřazen procesorový čas a je vykonáván kód
- 4) Zablokovaný / Blocked: Proces čeká na vyřízení požadavku, mezitím neblokuje procesor
- 5) Odložený / Swapped: Pokud procesy čekají moc dlouho tak se můžou z RAM odložit na disk na spešl partition
- 6) Zastavený / Zombie: Vyřazen z fronty na procesor, ještě nebude zahájeno mazání dat
- 7) Ukončený / Terminated: Požádáno o ukončení procesu, proces ještě existuje v paměti ale už nebude spuštěn

c) Synchronizace procesů

- Procesy můžeme synchronizovat pomocí zámky, semaforů
- Procesy také spolu můžou komunikovat přímo
 - Synchronně
 - Asynchronně
- Při komunikaci můžou využít
 - Sdílenou paměť - sdílejí si úsek RAM
 - Fronty zpráv - procesy si zasílají zprávy
 - Roura - zapisují si mezi sebou daty skrze "rouru"
 - Signály, socket - častá metoda, předem si nastaví zprávy, často standard POSIX

d) Vlákna

- Složitější programy lze rozdělit na nějaké dílčí na sobě navzájem nezávislé činnosti, které pracují paralelně jako malé procesy
- Sdílejí mezi sebou paměť
- Anglicky threads
- POSIX Threads, Windows Threads

e) Uvážnutí a souběh v OS

- Souběh
 - Dvě entity pracují se stejnými daty, bez toho aby spolupracovali
 - Procesy data náhodně mění a může dojít k poškození dat
- Řešení souběhu
 - Atomická operace - musí běžet celá, nerozdělená na dílčí části
 - Zámky - pokud někdo hrabe na data tak se to zamkne pro ostatní
 - Semafor - přístup několika "vozidel" na parkoviště, zámek je v podstatě "binární semafor"
- Uvážnutí (deadlock)
 - Procesy se navzájem zablokují, protože každý potřebuje to, co má ten druhý
- Řešení uvážnutí
 - Jednomu procesu ukradneme prostředky
 - Jeden proces zabijeme
 - Restartujeme systém (lol)

5) Paměť

a) Typy pamětí, rozdíl mezi operační pamětí a úložištěm, registry, cache

- Operační paměť - ukládají se na ni procesy, rychlé
- Úložiště - sklad paměti, pomalé
- DRAM - udržují elektrický náboj v buňce, tento náboj ztrácejí
- SRAM - statická, rychlejší ale drahá, nachází se v procesorech
- Registry - součást ALU, načítají se odsud data pro operace a výsledky operací
- Cache - rychlá vnitřní paměť v procesoru, softwarově se nedá ovládat

b) Virtualizace paměti

- Lhaní procesům o místech v paměti (praktické a bezpečné)
- Procesy pracují s virtuální adresou, reálnou zná jenom OS
- Potřeba podpory CPU (memory management unit - MMU)

c) Základní úkoly správy paměti v OS

- Každý proces dostane přidělenou svou vlastní část operační paměti
- Operační systém odpovídá za ochranu paměti
- Evidence volných a přidělených místech v paměti

d) Dynamické přidělování paměti

- Memory Allocator (součást jádra, nechráněný režim)
- Musí vědět jaká paměť je volná, komu patří paměť, jaká paměť je kterého procesu
- Proces
 - Je k dispozici dostatek volné paměti?
 - Vybrat úsek, jenž splňuje nároky
 - Aktualizovat informace o volné paměti
- Čtyři základní metody pro realizaci těchto kroků
 - Nejpřesnější (best fit)
 - Co největší (worst fit)
 - První vhodný (first fit)
 - Nejbližší vhodný (next fit)

e) Stránkování a další virtualizační metody

- Stránkování
 - Odkládání paměti do trvalého úložiště
 - Paměť je rozdělena na stejně velké úseky (stránky, pages)
 - Paměť RAM je rozdělena na rámce (frames), tzn. pro přístup musíme znát číslo stránky, zdali je odložena nebo aktivována v konkrétním rámci RAM

6) Úložiště a systémy souborů

a) Jaké znáte typy úložišť?

- Disky SSD - EEPROM, podobné RAM, rychlé, každá buňka má omezený počet zápisů
- Přenosné paměti flash - SSD v malé podobě s USB, dražší než SSD (10x)
- Karty SD - malé, omezený počet zápisů, drahé
- Optická média (CD, DVD, blu-ray) - levné 😊, omezená kapacita, pomalý zápis
- Magnetické pásky - zálohování, velká kapacita a dobrá rychlost, data vydrží dlouho, bohužel přestože bych si tam rád zálohoval tak mechaniky stojí 100k

b) Definujte pojem systém souborů a jeho důležité vlastnosti.

- Systém souborů =
 - uspořádání dat v úložišti
 - řeší vytváření souborů, složek, práce s nimi
 - technické parametry (struktura hlavičky souborů, rozdělení média, oprávnění k přístupů a další
- Soubor se obecně skládá z
 - Jméno a identifikátor
 - Popis umístění
 - Velikost
 - Časové informace
 - Ochrana před neoprávněným přístupem
- Základní funkce: vytvoření souboru, otevření souboru, přesun, kopírování

c) Uveďte příklady existujících FS a srovnajte je mezi sebou.

- FAT, FAT-16, FAT32 a další (zkratka pro: Fat Allocation Table)
 - Velmi starý a velmi jednoduchý
 - Systém neumožňuje oprávnění pro soubory
 - Clustery - dělení média (512B - 32kB) => FAT32 podporuje max 2^{32} clusterů (max. 16TB na médium)
 - kti v Microsoftu si patentovali nějaké části FAT a většina vypršela v roce 2011
- VFS (Linux Virtual File System)
 - Není to přímo systém souborů ale softwarová vrstva v OS Linux, kterou používá pro přístup k připojeným diskům, spravuje příkazy jako open()
- ext (Extended file system)
 - vyvíjen přímo pro OS Linux
 - V současnosti se používá ext4
 - teoreticky zvládne až 1 YiB, datумы v rozmezí: 1901-2446
 - systém chrání operace nad objekty žurnálem
- NTFS
 - kód není open source, patří Microsoftu
 - max. 256TB, datum 1601-60056
 - zajímavé schopnosti: aut. komprimace před uložením na disk, šifrování
 - Linux s tím už umí pracovat
- APFS (Apple File System) (2017)
 - macOS !!!!!
 - 1970-2554
 - Podporuje copy on write, šifrování dat, kontrolní součty metadat
- LittleFS
 - Společnost ARM
 - Zephyr, Mbed, thread-safe (mohou k němu přistupovat nezávislá vlákna kódu)
 - ochrana integrity dat
- ZFS - systém pro velká úložiště, 2010 změna licence na closed-source, nej. pro servery

7) Zařízení

a) Uveďte příklady zařízení (devices) z hlediska operačního systému.

- Zařízení = periférie, hardware komponenty připojené k PC a ovládaný OS
- Bloková zařízení
 - čtení (read) a zápis (write) dat
 - seek - pro vyhledávání
 - zařízení chápáno jako posloupnost bloků
 - disk
- Znaková zařízení
 - lze reprezentovat například textovým souborem
 - get - čtení souboru, put - zápis znaku, read, write po sekvencích
 - myš, klávesnice
- Síťová zařízení
 - mívají samostatné rozhraní OS
 - propojení realizováno sockety
- Hodiny a časovače
 - měří časový úsek
 - umí určit čas, nastavit hlídače na čas
- Ovladače

b) Jaké činnosti vykonává OS při správě zařízení? (Princip řadiče)

- inicializace - zavedení zařízení do OS
- připoj zařízení - plug & play
- ovládání zařízení - zápis a čtení, jiné I/O operace
- start I/O - k zahájení a ukončení přenosu dat
- obsluha přerušení
- zrušení operace, rychlý průchod, oznámení o vypnutí, záznam chyb

c) Vysvětlete princip řadiče (driver) zařízení.

- Speciální program
- Prostředník mezi zařízením a operačním systémem (požadavky se zprostředkují zde, taková API)
- a) detekce zařízení
- b) instalace ovladačů
- c) přidělení zdrojů (paměti, procesorový čas)
- d) správa energie (režim spánku)
- e) komunikace se zařízeními - pomocí API a ovladačů
- idk freestyle prostě

8) Embedded & RealTime

a) Srovnajte potřeby OS pro vestavěná zařízení s OS pro osobní počítače či servery.

b) Co je režim reálného času?

- Systém pracující v režimu reálného času musí být deterministický, bez nejistot, reprodukovatelný a umožňující kvalifikovaný odhad doby odezvy.
- Deterministický systém – při zpracování vnějších událostí dokáže zachytit jejich časovou posloupnost a časové odstupy okamžiků, v nichž události nastávaly. Determinističnost vyžaduje přesné měření času a fronty událostí.
- Musí zajistit zpracování procesů v definovaném pořadí.
- Implementace – tam, kde je potřeba řídit činnost nějakého zařízení nebo více zařízení synchronizovat.
- Požadujeme jistotu obsluhy událostí. Události musí být obslouženy vždy včas.
- Procesy mají deadline:
- Hard deadline se musí stihnout za každou cenu (kritické události, např. zabrzdění auta).
- Soft deadline se může prošvihnout (periodické, nekritické událost, např. čtení z meteostanice).

c) Uveďte příklady existujících embedded OS

d) Výhody a nevýhody Linuxu pro nasazení ve vestavných aplikacích.

9) Standardy

a) Standard POSIX: co definuje, proč existuje, k čemu je dobrý.

- POSIX = “Portable Operating System Interface” + (U)nix
- odvozeno z Unixu
- definuje: strukturu hlavičkových zdrojových kódů OS
- základní strukturu adresářů
- chybové kódy
- datové typy
- typy signálů
- vlákna, sockets, procesy
- seznam funkcí, jež tvoří API
- vlastnosti jakéhokoliv shellu a seznam příkazů
- certifikace: splnění pravidel POSIX posouzené certifikační autoritou (drahé, většinou na to kašlou a jen se prohlašují za POSIX, Linux certifikaci nemá)

b) Linux Standard Base: co definuje, proč existuje, k čemu je dobrý.

- společný projekt několika distribucí OS Linux
- zahrnuje definice pro: strukturu adresářů, strukturu bin. spustitelných soub., základní systémové knihovny, inicializace systému (init, cron, runlevels), práce s uživateli a skupinami, balíčkovací systémy

c) Oborové a průmyslové standardy: proč existují, na co se zaměřují, kdy se vyplatí certifikace

- ECSS: vesmír
- Automotive
 - ISO 26262
 - Misra standard
- The GNU Coding standard
 - vytisknout a spálit (pokud píšete Linux Kernel)
- Linux Kernel Coding Style
 - krátké řádky,

10) Start systému

a) Popište start systému od zapnutí po zahájení činnosti OS

- 1) Start hardware
 - Probouzí se procesor, popř. jiný řídicí obvod hlavní desky
 - Procesor si automaticky nahraje firmware do paměti RAM
 - Složitější systémy s velkými procesory buď mají v paměti ROM připojem startovací software nebo je na desce jiný, jednoduchý procesor, jehož úkolem je startovací program spustit a inicializovat procesor (zavaděč první fáze, first-stage boot loader)
 - Dříve se používal výhradně BIOS u x86, později se definoval standard UEFI (univerzální rozhraní pro startovací SW) a BIOS je na ústupu
 - Tudíž jsou autoři startovacího programu výrobci základních desek
 - Startovací médium - instalace je třeba z flashky nebo CD, po inst. pevný disk
 - UEFI obsahuje basic drivery pro komunikaci se zařízeními a sběrnicemi (třeba abych mohl OS z něčeho nainstalovat)
 - Poslední je načten seznam dostupných OS a zavádí se samotný OS (zavaděč druhé fáze)
- Dílčí části první fáze startu (UEFI)- fáze
 - Security Phase (SEC) - dočasná operační paměť než se připraví RAM (používá se často cache v procesoru), dnes se používají čipy TPM (trusted platform modules), které ukládají klíče pro kontrolu pravosti zavedeného software
 - Pre-EFI Initialization (PEI) - zavádění modulů pro inicializaci hardwaru (připojení hlavní paměti RAM a její základní konfiguraci, používá se vyrovnávací paměť procesoru)
 - Jakmile je RAM ready - spouští se řadiče zařízení (Driver Execution Environment) - inicializují se obvody základní desky a její periferie, nakonec se vybírá zařízení, z čeho se bude startovat systém
 - Transient System Load (TSL) - předává se řízení zavaděči OS

b) Co je zavaděč OS? Jaké znáte?

- Druhá fáze
- CPU pracuje v nechráněném režimu - nahrání na začátek paměti RAM
- Cíl: připojit sys. disk, nahrát do paměti jádro systému a předat řízení
- Nabídka pro uživatele - při několika OS
- GNU GRUB
 - Dominantní zavaděč OS Linux na osobních PC a serverech
 - Zobrazí nabídku systémů, předává jádru parametry, které se použijí při startu
 - Má vlastní příkazový řádek, je podobný příkazové řádce v Linuxu
 - Nastavení zavaděče je uloženo v souboru na systémovém disku
- Zavaděč Windows
 - a) Správce startu, b) Zavaděč systému, c) Zavaděč pro zotavení
 - Každá část je uložena v samost. spustitelném souboru na sys. disku v hl. složce
 - Win mají i skrytý disk. oddíl pro zotavení (cca. 500MB) s továrním nastavením
- U-Boot (malý bratr Linuxu)
 - Vestavěná zařízení
 - Zavaděč první i druhé fáze
 - Uložen v binární podobě je na začátku start. paměť. média
 - Je zaváděn do RAM a inicializuje hardware, spouští operační systém
 - Firmware třeba v ROM nahrává U-Boot nebo nějaký malý mikroprocesor
 - Atmel používá firmware bootstrap
 - Používá vlastní řadiče zařízení