Technologie OS I.

| 1. | OPE | OPERAČNÍ SYSTÉM (OS)2 | | | | |
|----|------------|--|----|--|--|--|
| | 1.1 | ÚČEL OS | .2 | | | |
| | | 1.1.1 Složky správy výpočetního systému | | | | |
| | 1.2 | FUNKCE OS. | | | | |
| | 1.3 | KOMPONENTY OS | | | | |
| | 1.4 | Rozhraní | .2 | | | |
| | | 1.4.1 Pro uživatele | | | | |
| | | 1.4.2 Pro aplikační programy | | | | |
| | 1.5 | PROCES | | | | |
| | | STRUKTURA OS | | | | |
| 2. | TYPY | OS Z HISTORICKÉHO HLEDISKA | .3 | | | |
| | 2.1 | Monitory | .3 | | | |
| | 2.2 | DÁVKOVÝ JEDNOUŽIVATELSKÝ SYSTÉM | | | | |
| | 2.3 | DÁVKOVÝ VÍCEÚLOHOVÝ SYSTÉM | | | | |
| | 2.4 | Systémy se sdílením času a interaktivním přístupem uživatele | | | | |
| | | 2.4.1 Rozdíly oproti ostatním typům | | | | |
| | | OPERAČNÍ SYSTÉM 360 | | | | |
| 3. | ZAŘÍ | ZENÍ VSTUPU A VÝSTUPU | .4 | | | |
| | 3.1 | ZNAKOVĚ ORIENTOVANÁ | .4 | | | |
| | 3.2 | BLOKOVĚ ORIENTOVANÁ | | | | |
| | 3.3 | ZAŘÍZENÍ V/V SE OBECNĚ SKLÁDÁ ZE DVOU ČÁSTÍ: | .4 | | | |
| | | 3.3.1 Kontrolér | | | | |
| | | Registry kontroléru | | | | |
| | 3.4 | ZPŮSOBY PŘÍSTUPU CPU K REGISTRŮM ZAŘÍZENÍ VSTUPU-VÝSTUPU | | | | |
| | | 3.4.1 Výhody a nevýhody | | | | |
| | 3.5 | METODY PŘESUNU DAT | .5 | | | |
| | | 3.5.1 Programová obsluha | | | | |
| | | 3.5.2 Obsluha pomocí přerušení | | | | |
| | | 3.5.3 Přímý přístup do paměti | | | | |
| 4. | MS-I | DOS | .6 | | | |
| | 4.1 | ZÁKLADNÍ CHARAKTERISTIKA | .6 | | | |
| | | VERZE MS-DOS | | | | |
| | 4.3 | ZÁKLADNÍ RYSY | | | | |
| | | 4.3.1 Interpret přídazů | | | | |
| | | ZAVADĚČ SYSTÉMU | | | | |
| | 4.5 4.6 | MSDOS.SYS | | | | |
| | 4.7 | COMMAND.COM | | | | |
| | | 4.7.1 Práce command.com | - | | | |
| | | 4.7.2 Části Command.com | | | | |
| | 4.8 | SPRÁVA PAMĚTI | | | | |
| | | Souborový systém | | | | |
| | 4.10 | PÁR PŘÍKAZŮ | .8 | | | |

1. Operační systém (OS)

Je základní programové vybavení počítače, nezbytné pro jeho provoz

1.1 Účel OS

Správa a poskytování služeb výpočetního systému

1.1.1 Složky správy výpočetního systému

- Hardwarově závislá část jádro
- HW nezávislá část uživatelská část

1.2 Funkce OS

- Poskytování služeb aplikačnímu softwarovému vybavení (API aplikační rozhraní systému)
- Principem poskytování služeb je odstínění aplikací od rutinních, standardních činností (například otevření souboru, přidělení paměti, atd..)

1.3 Komponenty OS

- Správa procesů
- Správa operační paměti
- Správa souborového systému
- Správa periférií

1.4 Rozhraní

1.4.1 Pro uživatele

Umožňuje řízení systému

Textový režim

- Uživatel s OS komunikuje pomocí textových příkazů.
- Nevýhoda: uživatel musí znát příkazy, aby mohl se systémem pracovat
- Výhoda: Uživatel přesně ví, co právě OS dělá (má ho pod kontrolou)

Grafický režim - GUI

- Komunikace s OS probíhá pomocí interaktivních grafických prvků (menu)
- Nevýhoda: Uživatel nemá přehled a práci OS
- Výhoda: Uživatel nemusí znát příkazy, vybírá si pouze z nabízených možností

Dnešní OS mají převážně obě rozhraní – textová (ve Windows je to příkazová řádka) i grafická.

1.4.2 Pro aplikační programy

- Aplikační programový interface (API)
- Zpřístupňuje služby OS ostatním programům

1.5 Proces

- Klíčový element OS
- Posloupnost událostí definována svým účelem nebo účinkem, uskutečňovaná za daných podmínek
- Proces je vykonávající se program (program vykonávaný v daném kontextu, prakticky totéž co úloha, při přepínání úloh)

- V systému musí existovat vhodný signál, který pozastaví nebo znovu aktivuje příslušný proces.
 Dojde-li k pozastavení procesu, musí existovat nějaký mechanizmus uložení kontextu procesu a jeho zpětné obnovení při reaktivaci procesu.
- Každý proces má unikátní identifikátor (zvláště ve více úlohovém nebo víceuživatelském prostředí). Většinou je to binární číslo (16 nebo 32 bitů)
- Proces může zrodit další proces (procesy). Což znamená, že procesy mají hierarchickou strukturu (rodič – potomek)

1.6 Struktura OS

- o zpočátku měly OS strukturu jako libovolný jiný program (the big mess)
- o Později se přešlo na vrstevnatou strukturu (Dijkstra)
- Následovala koncepce virtuálního stroje (OS 3700)
- Vývoj postupně vytlačoval kód do vyšších vrstev OS, což vedlo k minimalizaci jádra OS (kernel)
- Výsledkem je model OS typu klient server

Rozdělení OS na relativně autonomní části (servery), spravující plně určitou činnost a nemajíce dostup k hardware znamenitě vylepšilo spolehlivost a přenositelnost.

2. Typy OS z historického hlediska

2.1 Monitory

- Jednoduché programy
- o Umožňují zavádět do počítače aplikační programy v binární formě a popřípadě je také odlaďovat

2.2 Dávkový jednouživatelský systém

- Zpracovává úlohy (batch, processing) pomocí souboru příkazů definující zpracování zakázky
 v některém skriptovacím jazyce pro řízení úloh (OS 360, JCL Job Control Language)
- Základní výhodou i nevýhodou tohoto typu OS je nepřítomnost aplikačního programátora při zpracování úlohy.

2.3 Dávkový víceúlohový systém

- Umožňuje zpracování více úloh takřka současně
- Základním cílem těchto OS je odstranění prostojů procesu (nejdražší části počítače) při dávkovém zpracování úloh
- Podstatou je systém s několika úlohami a přepínáním kontextu podle připravenosti úlohy (a dostupnosti zdrojů).
- Řízení systémů v reálném čase řízení technologických procesů (především vojenská zařízení)
 Klíčovým parametrem těchto OS je doba odezvy a spolehlivost.

2.4 Systémy se sdílením času a interaktivním přístupem uživatele

- Je to OS pracující v reálném čase
- Je schopný zpracovat pomocí mechanizmu sdílení času více úloh najednou, který umožňuje bezprostřední (interaktivní) komunikaci s uživatelem (nebo i uživateli)
- Většina moderních univerzálních OS vychází z této koncepce.

2.4.1 Rozdíly oproti ostatním typům

Interaktivně komunikuje s uživatelem – musí odpovědět do určitého času

2.5 Operační systém 360

- Umožňoval poprvé v historii běh několika úloh současně
- Nejprve pevný, později proměnný počet úloh
- TSO, vzdálené vstupy z terminálů
- Programové vybavení pro řízení práce stroje JCL a pro vývoj aplikací: asembler, ALGOL, FORTRAN,
 COBOL RPG a PC/1

Fundamentální práce v oblasti OS a programování vůbec jsou spojeny s panem **Edgarem W. Dijkstrou** (Holanďan, **OS T. H. E**, zhruba rok 1968)

3. Zařízení Vstupu a Výstupu

- o OS také vždy definuje standardní zařízení pro vstup a výstup
- o Zařízení V/V jsou často reprezentována jako speciální soubory
- o Komunikaci mezi procesy zajišťuje speciální druh souborů zvaných roury (pipe)
- Zařízení V/V můžeme rozdělit do 2 základních skupin

3.1 Znakově orientovaná

- Data se přenášejí po znacích (character devices)
- Nejmenší adresovatelnou jednotkou je jeden znak (bajt)
- Klávesnice, sériový komunikační adaptér...

3.2 Blokově orientovaná

- Data se přenášejí po blocích (block devices)
- Nejmenší adresovatelnou jednotkou je jeden blok (řada bajtů)
- o Disková nebo pásková jednotka...

Existují samozřejmě i zařízení, které se nedají zařadit ani do jedné z uvedených kategorií (například časovač nebo hodiny reálného času). Nicméně, tento přístup umožňuje značně zjednodušit správu zařízení V/V.

| zařízení V/V | rychlost (sec -1) |
|-------------------|-------------------|
| klávesnice | 20 bajtů |
| myš | 100 bajtů |
| 56k modem | 7 KB |
| laserová tiskárna | 100 KB |
| IDE disk | 4 MB |
| 40x CD-ROM | 6 MB |
| Fast Ethernet | 12.5 MB |
| EIDE ATA-2 | 16.7 MB |
| Firewire | 50 MB |
| Gigabit Ethernet | 125 MB |
| PCI Bus | 528 MB |

3.3 Zařízení V/V se obecně skládá ze dvou částí:

- Mechanické
- Elektronické

Modulární členění na řídící kontrolér, které obsahuje obecnou části řídící elektroniky a mechaniku (s nezbytnou speciální elektronikou) je užitečné, protože umožňuje spravovat podobné zařízení stejným způsobem. Rozhraní mezi kontrolérem a mechanikou je (takřka vždy) standardizováno (EIDE, PCI). Takové řešení zhusta dává

příležitost připojit k jednomu kontroléru několik (běžně 2-8) mechanik. Nezanedbatelnou výhodou je i láce (levnost) produktů.

3.3.1 Kontrolér

- Je řídící jednotka zařízení vstupu-výstupu
- Rozhraní mezi kontrolérem a mechanikou zařízení většinou pracuje na velmi nízké úrovni.
 Příklad u pevného disku, který je klasicky formátován (sektor 512 bajtů) se při operaci čtení přenáší sériový proud dat o délce 4568 bitů hlavička (preamble), data a kontrolní suma (ECC).

Úkolem kontroléru

o je konvergovat proud bitů na bajty, které uloží do vyrovnávací paměti

Zkontroluje, pomocí ECC, zda byly data přeneseny bezchybně. Pokud došlo k chybě, pokusí se jí opravit. Následně dodá data a zprávu o výsledku operace (nebo oznámí selhání).

Registry kontroléru

- o Každý kontrolér potřebuje registry pro komunikaci s nadřízenou jednotku (CPU)
- Většinou se jedná o registr datový, řídící, stavový
- o Datový registr obvykle bývá jeden (pro čtení i zápis), řídích a stavových může být vícero

3.4 Způsoby přístupu CPU k registrům zařízení vstupu-výstupu

- o 1. Registry jsou přímo mapovány do adresového prostoru operační paměti (DEC, PDP-11)
- 2. Registry mají vlastní adresový prostor, do kterého se přistupuje pomocí příslušeného typu instrukcí (IBM, OS360)
- Existuje i kombinované řešení datové registry jsou mapovány do operační paměti a řídící a stavové registry mají samostatný adresový prostor a příslušné instrukce pro komunikaci (Intel Pentium – datové registry v oblasti od 640 KB – 1 MB)

3.4.1 Výhody a nevýhody

- Některé programovací jazyky (C, C++) nemají instrukce vstupu-výstupu, takže je nutné napsat část kódu v asembleru. To je zjevná nevýhoda
- Základní výhodou registrů mapovaných přímo do operační paměti je, že s nimi lze pracovat jako s libovolnou jinou proměnnou. Což znamená, že mohu použít stejné instrukce i stejné mechanizmy řízení přístupu.

Příklad:

Pokud potřebuji otestovat bit ve stavovém registru, použijí instrukci TEST. V případě, že registr není přímo mapován v operační paměti, musím nejprve přečíst (=přenést do paměti) obsah registru instrukci IN a poté provést testování bitů. Dosti nepříjemným nedostatkem registrů mapovaných do operační paměti je nutnost prohledat značně větší adresový prostor při hledání určitého zařízení. Bohužel, u soudobých procesorů má tato metody a některé další zásadní nedostatky. Především, sběrnice paměti je optimalizována na rychlé přenosy (zařízení V/V jsou většinou pomalá). Dále, paměť cache je třeba pro příslušný úsek paměti, kam jsou mapovány registry vstupu a výstupu selektivně vypnout.

3.5 Metody přesunu dat

- Známe tři základní metody pro přesun dat mezi procesorem a zařízením vstupu-výstupu
- Každá metoda má své klady a zápory
- Aby věc byla komplikovanější, v důsledku evoluce se priority přesouvají

3.5.1 Programová obsluha

Představuje nejjednodušší variantu přenosu dat

Je vhodná pro jednotky, které pracují přibližně stejnou rychlostí

Postup

- o Řídící jednotka vyšle požadavek a v čekací smyčce čeká na jeho splnění
- Podřízená jednotka provede žádané (nebo neprovede, pokud z nějakého důvodu nemůže) a oznámí, nastavením příslušných bitů ve stavovém registru svůj stav (nebo důvod neprovedení)

3.5.2 Obsluha pomocí přerušení

- Je poněkud komplikovanější
- Tento způsob obsluhy je zhusta časově méně náročná než programová obsluha (úměrně rozdílu rychlosti práce jednotek)
- Jistou komplikací je nutnost nejprve iniciovat (nastavit) přerušovací systém
- Naopak vítanou výhodou může být fakt, že podřízená jednotka může být schopna (je-li tak konstruována) aktivně přivolat pozornost nadřízené jednotky (pokud to potřebuje).
- Začátek operace je stejný jako v předchozím případě

Postup

- Řídící jednotka vyšle požadavek, ale nečeká na jeho splnění a pokračuje v činnosti (tedy, může-li, má-li jinou práci). Když podřízená jednotka splní požadované, vyvolá přerušení
- Rutina obsluhy přerušení provede potřebné (třeba přesune data z/do vyrovnávací paměti) a signalizuje ukončení operace

3.5.3 Přímý přístup do paměti

- Představuje nejelegantnější, ale také nejkomplexnější způsob datových přenosů mezi zařízením
 V/V a procesorem
- Základním předpokladem je existence kontroléru přímého přístupu do paměti (DMA)
- Ten může být součástí základní desky (nejčastěji) nebo samotné periferie
- Přenáší se vždy blok dat (lze přenést i jeden bajt,postrádá to však smysl)

Postup

- Operace je zahájena naprogramováním kontroléru: kolik se má přenést, odkud a kam
- Další činnost už řídí kontrolér
- Přenos se v zásadě realizuje dvěma způsoby:
 - řízení nad sběrnicí převezme kontrolér vždy jen na přenos jednoho slova metoda kradení cyklů (cycle stealing) – procesor se o sběrnici dělí s kontrolérem DMA
 - při druhém způsobu převezme kontrolér DMA řízení nad sběrnicí na celou dobu přenosu (bloku) dat. Blok dat se přenese "najednou", procesor je odstaven od sběrnice na delší dobu, což může způsobit jisté problémy (burst mode shlukový přenos, také intervalový)

4. MS-DOS

- o Nejrozšířenější OS firmy Microsoft, je to první operační systém určený pro jednoduchou obsluhu
- Vydán v roce 1981

4.1 Základní charakteristika

- Je jednouživatelský
- Je schopen používat informace pouze z jednoho vstupního zařízení
- Je jednoprogramový, podporuje spuštění a běh pouze jedné úlohy (výjimku tvoří služební programy PRINT)

 Tvoří hierarchickou strukturu - systém je rozdělen do několika částí, mezi kterými je definováno rozhraní, jehož prostřednictvím mezi sebou jednotlivé části komunikují

4.2 Verze MS-DOS

| Verze | Popis |
|---------------|--|
| MS-DOS v1.1 | Rok 1981, první vydání DOS |
| MS-DOS v2.1 | Rok 1983, IBM XT, nové příkazy, podpora HDD (5 MB) |
| MS-DOS v3.2 | Duben 1986, Podpora sítě, podpora 3,5 disket 720 KB |
| MS-DOS v3.3 | Rok vydání 1987, podpora 3,5 disket 1,44MB a PS/2 |
| MS-DOS v4.0.1 | Rok vydání 1989, první verze manažera DOS Shell |
| MS-DOS v5.0 | Rok vydání 1991, prvně použit HIMEM.SYS |
| MS-DOS v6.22 | Rok vydání 1994, poslední oficiální a samostatný DOS |
| MS-DOS v7.0 | Součástí Windows 95 |
| MS-DOS v7.1 | Součástí Windows 98 |
| MS-DOS v8.0 | již velmi "ořezaná" – součástí Windows ME |

4.3 Základní rysy

o Po spuštění PC je řízení předáno interpretu příkazů

4.3.1 Interpret přídazů

- o Interpret příkazů se skládá ze tří části:
 - Obsluha přerušení, kritická chybová hlášení, rutiny pro spuštění procesu typu potomek
 - Rutiny pro zpracování dávkových souborů a procedury přidělování paměti pro zavádění programů
 - Realizace interpretu příkazů OS (provádí veškeré interní příkazy)

4.4 Zavaděč systému

 Je umístěn v prvním sektoru pevného disku, po načtení zkontroluje, zda jsou na disku soubory IO.SYS a MSDOS.SYS v uvedeném pořadí. Pokud soubory nenajde, prohledává ještě další disk, a při nenalezení vydá chybovou zprávu. Jestliže je najde, předá řízení io.sys.

4.5 IO.SYS

- Je programovým rozšířením modulu BIOS a při práci s operačním systémem je trvale umístěn v paměti RAM. Slouží k provádění tří základních činností:
- Tvoří nadstavbu BIOS, zajišťující co nejefektivnější provádění I/O operací.
- Slouží k odstranění chyb v BIOSu, změny přímo v BIOSu by se musely provést výměnou obvodů v paměti ROM, proto je možno změnit adresy přerušení tak, aby přerušení místo do BIOS vedlo do IO.SYS.
- Slouží pro zajištění obsluhy dalších přídavných zařízení doplněných do sestavy počítače.

4.6 MSDOS.SYS

 Zajišťuje správu systémových prostředků počítače, tj. operační paměti, v/v zařízení a systému ovládání souborů. Při práci s operačním systémem je trvale rezidentní v paměti RAM.

4.7 COMMAND.COM

• Tvoří nejvyšší vrstvu operačního systému a zajišťuje komunikaci uživatele s operačním systémem nebo s jeho vyššími úrovněmi. Operačním systémům je realizován procesorem příkazů uloženým ve formě souborů command.com na systémovém disku. Tento soubor je umístěn za io.sys a msdos.sys. Hlavní činností command.com je čtení příkazů, které vkládáme z klávesnice a jejich provedení. Dále zajišťuje automatické provádění posloupnosti příkazů, uložených do tzv. dávkového souboru (ten má formát textového souboru ASCII). Zvláštním případem příkazového souboru je autoexec.bat.

4.7.1 Práce command.com

 vložíme-li do operačního systému příkaz, zkontroluje, zda se jedná o vnitřní příkaz operačního systému a tento sám provede. V druhém případě se jedná o vnější příkaz, command.com jej vyhledá postupně s rozšířením extenzí .com, .exe, .bat.

Najde-li soubor .com nebo .exe, zavede jej do RAM a předá mu řízení počítače.

Najde-li soubor .bat, provede automaticky danou posloupnost příkazů operačního systému. Disk s daným dávkovým souborem musí být během práce k dispozici, neboť command.com z něj čte příkazy postupně.

4.7.2 Části Command.com

- První část je trvale uložena v paměti RAM, obsahuje rutiny pro obsluhu chyb operačního systému (zobrazení chybových zpráv)
- Druhá část se ukládá do paměti pouze dočasně při zavádění operačního systému, označuje se jako inicializační. Zjišťuje, zda na disku, z něhož je systém zaváděn, je umístěn autoexec.bat. Pokud ano, zajistí jeho provedení.
- Třetí část je nejrozsáhlejší, obsahuje vlastní standardní procesor příkazů (včetně rutin pro provádění vnitřních příkazů). Rutinu pro zavádění do paměti RAM a provádění vnějších příkazů s rozšířením .com a .exe. Tato část vytváří na obrazovce přihlášení operačního systému. Při opětovném zahájení práce command.com se zjistí, zda je tato třetí část v paměti RAM a pokud není, zavede jej z disku.

4.8 Správa paměti

- o Paměť je v zásadě omezena na 640 kB lze obejít pomocí driverů paměti
- Přístupové segmenty 64 kB
- Paměť přidělována v paragrafech (16 B)

4.9 Souborový systém

- Hierarchické uspořádání
- Lokátor FAT
- Názvy o délce 8+3 znaků ber rozlišení velkých a malých písem

4.10 Pár příkazů

| Příkaz | Co dělá |
|--------|-------------------------------------|
| CD | Změna adresáře |
| DIR | Výpis obsahu adresáře |
| COPY | Kopírování souborů |
| XCOPY | Kopírování adresářů (včetně obsahu) |
| MD | Vytvoření adresáře |

| Příkaz | Co dělá |
|--------|--------------------------------|
| RD | Zrušení adresáře |
| DEL | Zrušení souboru |
| REN | Změna jména (přípony) souboru |
| TYPE | Výpis obsahu diskového souboru |
| FORMAT | Formátování (FORMAT C:) |