



Univerzita Hradec Králové
Fakulta informatiky a managementu

System souborů

Mgr. Josef Horálek



- = Pro většinu uživatelů je systém souborů nejviditelnější součástí operačního systému
 - = provádí mechanismy pro on-line ukládání a přístup k programům a datům:
 - = operačního systému;
 - = všech uživatelů výpočetního systému;

- = System souborů se skládá ze dvou oddělených částí:
 - = z množiny souborů;
 - = ve kterých jsou uložena data;
 - = adresářové struktury;
 - = organizuje všechny soubory v systému;
 - = podává o nich informace;

- = Soubor je pojmenován pro pohodlí uživatelů
 - = ti se na něj odvolávají prostřednictvím jeho jména;
 - = jméno souborů je většinou řetězec znaků;
 - = některé systémy rozlišují mezi velkými a malými písmeny ve jménech souborů, zatímco jiné je přijímají jako ekvivalentní;
 - = je-li soubor pojmenovaný;
 - = stává se nezávislý na procesu;
 - = stává se nezávislý na uživateli;
 - = stává se nezávislý na systému, který ho vytvořil;

- = Soubor má v různých systémech různé atributy, typicky:
 - = jméno;
 - = Symbolické jméno souboru v lidsky čitelné podobě;
 - = typ;
 - = pro systémy, které podporují různé typy souborů;
 - = lokace;
 - = ukazatel na zařízení a umístění souboru na tomto zařízení;
 - = velikost
 - = součástí tohoto atributu je aktuální velikost souboru (v bytech, slovech nebo blocích)
 - = a jeho maximální možná velikost;

- = Soubor má v různých systémech různé atributy, typicky:
 - = ochrana
 - = informace o ochraně přístupu k souboru definují, kdo ho může číst, kdo spouštět, kdo zapisovat atd;
 - = datum, čas a uživatelská identifikace
 - = tyto informace mohou být uloženy:
 - = po vytvoření;
 - = poslední modifikaci;
 - = posledním užití souboru;

- = Informace o všech souborech jsou uloženy v adresářové struktuře vytvořené na odkládacím zařízení
 - = tyto záznamy mohou zabírat od 16 po více než 1000 bytů na každý soubor;
 - = v systémech s mnoha soubory může velikost adresáře dorůstat megabytových rozměrů;
 - = protože adresáře a soubory, musejí být energeticky nezávislé, jsou tedy uloženy na odkládacím zařízení a do paměti zaváděny část co části, jak je třeba;.

- = Soubor je abstraktní datový typ
 - = pro přesnou definici souboru musíme zavést operace, které mohou být se soubory prováděny;
 - = operační systém provádí systémová volání pro:
 - = vytvoření;
 - = k vytvoření souboru jsou nutné dva kroky
 - = nejdříve je třeba najít pro soubor dostatečně velký prostor na odkládacím zařízení;
 - = musí být vytvořen záznam pro nový soubor v adresáři
 - = do tohoto záznamu, kde je uloženo jméno souboru a jeho uložení v systému souborů;

- = operační systém provádí systémová volání pro:
- = zápis;
 - = k zápisu souboru je prováděno systémové volání, které vyžaduje specifikaci jména souboru;
 - = informace, která má být do souboru uložena;
 - = po přijetí jména souboru systém vyhledá adresář a v něm informaci o umístění souboru;
 - = systém musí uložit ukazatel zápisu na místo v souboru, kde má být zápis proveden;
 - = ukazatel zápisu musí být správně nastaven kdykoli má dojít k zápisu do nějakého souboru;

- = operační systém provádí systémová volání pro:
- = čtení;
 - = ke čtení souboru je prováděno systémové volání;
 - = vyžaduje specifikaci jména souboru;
 - = informaci o tom, kde v paměti je uložen následující blok souboru;
 - = je vyhledán patřičný adresář a v něm patřičný záznam, podle něj je nastaven ukazatel čtení na místo, od kterého má být provedeno následující čtení;
 - = kdykoli má nastat čtení souboru, je aktualizována pozice ukazatele čtení;
 - = protože většinou je soubor buď čten, nebo zapisován, mnoho systémů užívá pouze jeden ukazatel aktuální pozice;
 - = obě operace, čtení i zápis, užívají týž ukazatel, čímž je šetřen prostor a redukována složitost systému;

- = operační systém provádí systémová volání pro:
- = přemístění v souboru;
 - = je vyhledán adresář a v něm patřičný záznam;
 - = ukazatel aktuální pozice je nastaven na patřičnou hodnotu;
 - = přemístění pozice příští vstupní nebo výstupní operace v souboru nepotřebuje volat žádnou I/O operaci;
 - = tato akce je známa jako seek;
- = smazání;
 - = je nutné vyhledat adresář, ve kterém soubor leží;
 - = uvolnit celý úložný prostor, který byl souboru přiřazen;
 - = smaže se položka záznam o daném souboru v adresáři;
- = vypuštění souboru;
 - = uživatel chce ponechat nastavené všechny atributy souboru a pouze vyprázdnit jeho obsah;
 - = tato funkce umožňuje zachovat všechny atributy souboru (s výjimkou velikosti), která je nastavena na nulu;

- = Těchto šest základních operací představuje minimální sadu nutných instrukcí
 - = další potřebné příkazy;
 - = připojení nových dat na konec souboru (append);
 - = přejmenování souboru (rename);
 - = potřebujeme také sadu operací, které umožní uživateli číst a nastavovat atributy souborů;
 - = operace pro zjištění velikosti souboru;
 - = operace dovolující nastavit vlastníka souboru;

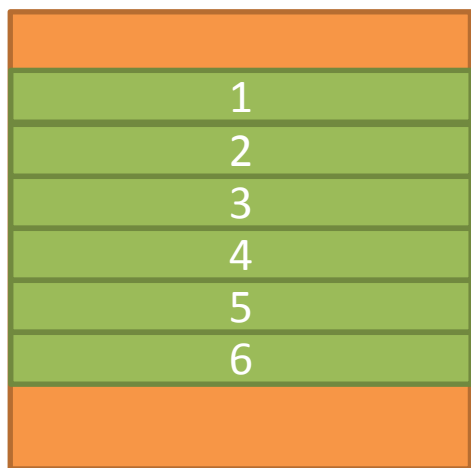
- = Většina operací se soubory v sobě zahrnuje vyhledání patřičného záznamu v adresáři
 - = některé OS jako první operaci se souborem implementují operaci open;
 - = OS si pak vede jen malou tabulku obsahující informace o všech otevřených souborech (Open-File Table);
 - = jestliže je potom prováděna další operace s otevřeným souborem, užije se index ukazující do Open_File Table a neprovádí se žádné vyhledávání;
 - = pokud již nebude soubor dále aktivně využíván, proces užije operaci close a OS vymaže záznam o souboru v tabulce otevřených souborů;
 - = některé systémy provádí operaci open automaticky, když je soubor poprvé použit;
 - = operace close je potom užita v okamžiku ukončení procesu, díky kterému byla vyvolána operace open;

- = Je-li soubor otevřen nějakým procesem a jiný proces na něj provede volání open, je přidána další položka do tabulky per-proces s ukazatelem na soubor a do systémové tabulky otevřených souborů;
 - = systém tedy udržuje nejrůznější informace asociované s otevřenými soubory:
 - = ukazatel do souboru (File pointer);
 - = čítač otevření;
 - = umístění souborů na disku;

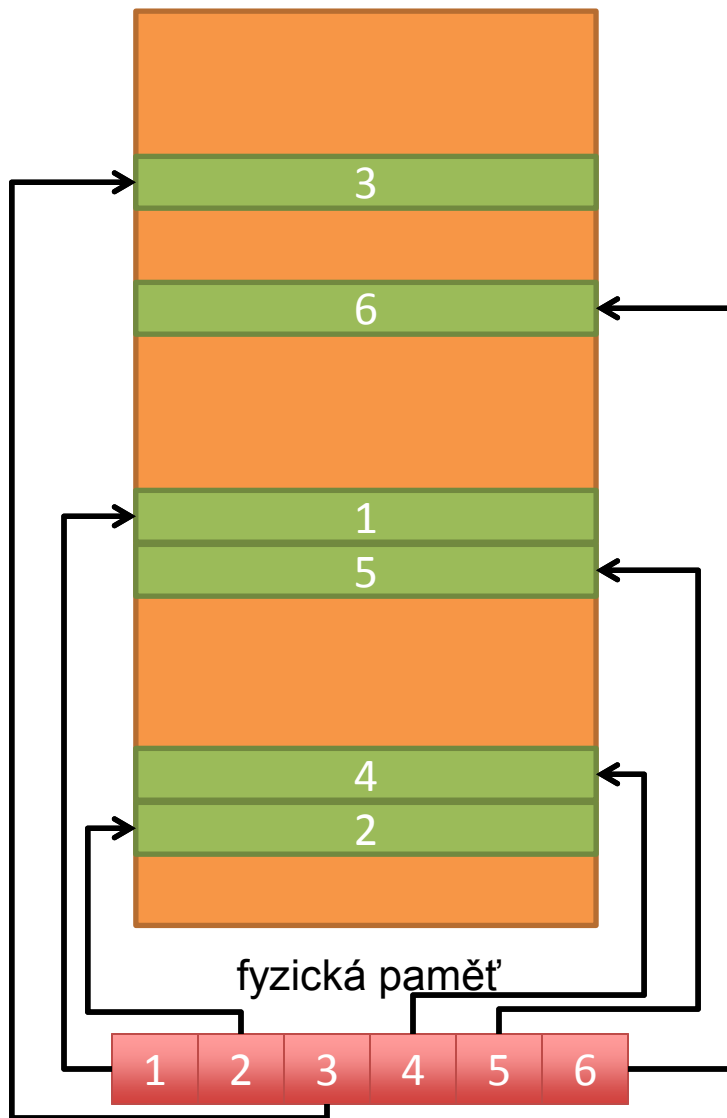
- = Některé OS provádí mechanismy pro víceprocesový přístup k souboru, aby mohli být sekce souboru sdíleny více procesy
 - = mapují jednotlivé sekce souboru do virtuální paměti (memory mapping);
 - = dovoluje asociovat část adresového prostoru virtuální paměti se sekcemi souboru;
 - = čtení a zápis do této oblasti paměti představuje čtení a zápis do souboru, což značně ulehčuje užití souboru;
 - = uzavření souboru vyvolá uložení všech virtuálně namapovaných sekcí souborů zpět na disk a vymazání souboru z virtuální paměti procesu;

= Zápis do souboru

- = libovolným procesem modifikuje sdílená data ve virtuální paměti;
 - = může ho provést libovolný proces, který má namapované sekce sdíleného souboru do své virtuální paměti;
 - = virtuální paměť každého procesu, který sdílí soubor obsahuje odkazy na stejné stránky fyzické paměti;
 - = na stránky, ve kterých je uložena kopie sdíleného souboru;
 - = přístup procesu ke sdílenému souboru je samozřejmě koordinován systémem, aby byl zajištěn vzájemně jedinečný přístup k souboru;

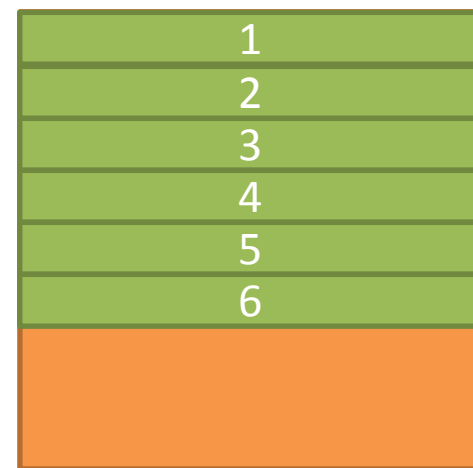


virtuální paměťový prostor procesu A



fyzická paměť

diskový prostor



virtuální paměťový prostor procesu A

- = OS rozeznává typy souboru, může se soubory lépe pracovat
 - = nejčastější technika implementace typu souboru je:
 - = vložit identifikátor typu jako součást jména souboru;
 - = jméno se potom skládá ze dvou částí:
 - = jména a přípony oddělených tečkou;
 - = typy souborů mohou být užity také k identifikaci jejich vnitřní struktury;

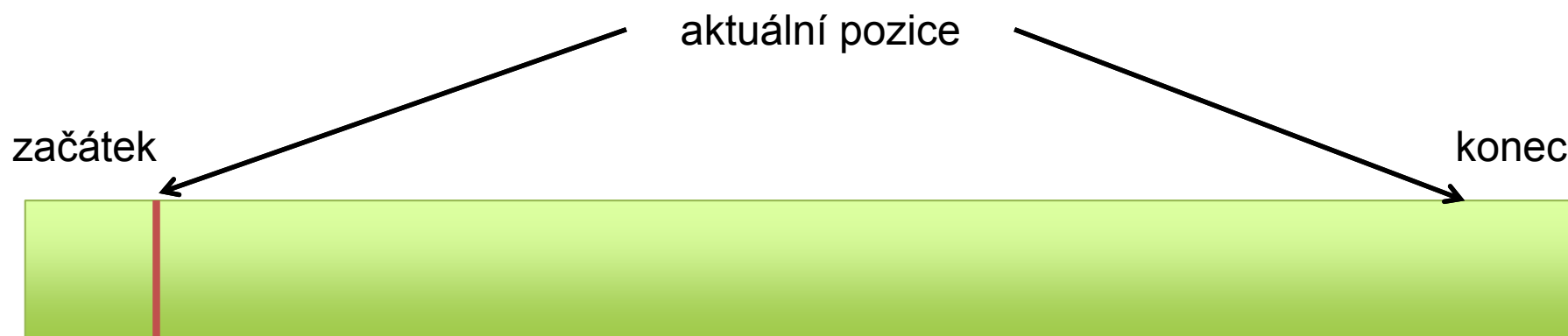
- = Některé OS využívají množinu speciálních operací, které se soubory vybraných datových typů manipulují
 - = tato myšlenka má nevýhodu v těžkopádnosti a přílišné velikosti kódu OS;
 - = pro každou podporovanou strukturu souboru musí OS obsahovat kód pro práci s ní;
 - = podporuje-li OS kompletně všechny struktury všech souborů;
 - = nová aplikace vytvořená pod takovýmto OS přinese problémy v případě, že bude vyžadovat organizaci dat v souboru způsobem, který OS nezná a nepodporuje;

- = Najít interně nějaký offset uvnitř souboru může být pro OS komplikované
 - = všechny diskové I/O operace pracují v jednotkách jednoho bloku (fyzický záznam);
 - = všechny bloky jsou stejné velikosti;
 - = je nepravděpodobné, že velikost fyzického záznamu bude přesně stejná jako velikost požadovaného logického záznamu;
 - = logické záznamy jsou proměnné délky
 - = Packing uložení logických záznamů do fyzických je řešením tohoto problému;

- = Délka logického záznamu, délka fyzického záznamu a technika packingu
 - = určují kolik logických záznamů se vejde do jednoho fyzického bloku;
 - = uložení může provést buď uživatelský program, nebo operační systém;
 - = nebo může být soubor definován jako sled bloků;
 - = všechny základní I/O operace pracují na těchto blocích;
 - = konverze logického záznamu do fyzického bloku je potom softwarový problém;

- = Soubory uchovávají informace
 - = má-li být informace použita, musí být přistoupeno k souboru, který ji obsahuje a tento soubor musí být zaveden do paměti;
 - = sekvenční přístup;
 - = přímý přístup;

- = Sekvenční přístup je nejjednodušší metoda přístupu
 - = informace v souboru jsou zpracovávány jeden záznam za druhým;
 - = velice častý přístup;
 - = operace prováděné na souboru jsou čtení a zápis;
 - = operace čtení čte následující část souboru a automaticky posune ukazatel následující I/O operace;
 - = stejně append, operace zápis provede zápis informací na konec souboru a automaticky posune ukazatel následující I/O operace na nový konec souboru;
 - = v některých systémech program může přeskočit dopředu nebo dozadu n záznamů, pro nějaké celočíselné;



- = Přímý přístup (nebo relativní přístup)
 - = soubor sestává z logických záznamů pevné délky;
 - = umožňuje tak programu číst nebo zapisovat záznamy rychle, ne v částečném pořadí;
 - = je založena na diskovém modelu souboru, protože disky umožňují náhodný přístup k libovolnému bloku souboru;
 - = při přímém přístupu je soubor představován očíslovanou sekvencí bloku nebo záznamu;
 - = přímý přístup k souboru umožňuje číst nebo zapisovat libovolný blok souboru;
 - = soubory s přímým přístupem umožňují rychlý přístup k velkému množství informací;

- = Operace se soubory musí být modifikovány tak, aby obsahovaly číslo bloku jako svůj parametr
 - = užíváme funkci `read n`
 - = `n` je číslo načítaného bloku
 - = číslo bloku, které používá uživatel v interakci s OS se nazývá relativní číslo bloku;
 - = je-li logicky záznam délky D , potom žádost o záznam N je převeden na I/O žádost o D bytu od místa $D \cdot (N-1)$ od začátku souboru;
 - = logické záznamy jsou pevné délky, je velmi jednoduché zapsat, přečíst nebo vymazat záznam;

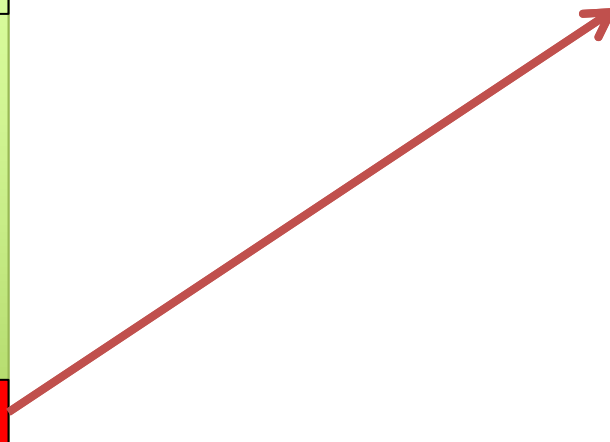
- = Další metody přístupu k souboru mohou být vytvořeny na základě metody přímého přístupu
 - = tyto metody většinou spočívají v konstrukci indexu pro každý soubor
 - = index, stejně jako rejstřík na konci knihy obsahuje ukazatele na různé bloky;
 - = pro nalezení nějaké položky v souboru nejdříve prohledáme index a potom užijeme ukazatel pro přímý přístup vedoucí k nalezení položky;
 - = u velkých souborů, může být i sám index příliš velký na to, aby byl uchovávan v paměti;
 - = možným řešením je vytvoření indexového souboru;
 - = primární indexový soubor potom obsahuje ukazatele do sekundárního indexu v souboru s daty;

indexový soubor

Ampère
Archimédes
Bell
Kepler

relativní soubor

Kepler Johannes, (1571-1630), ...

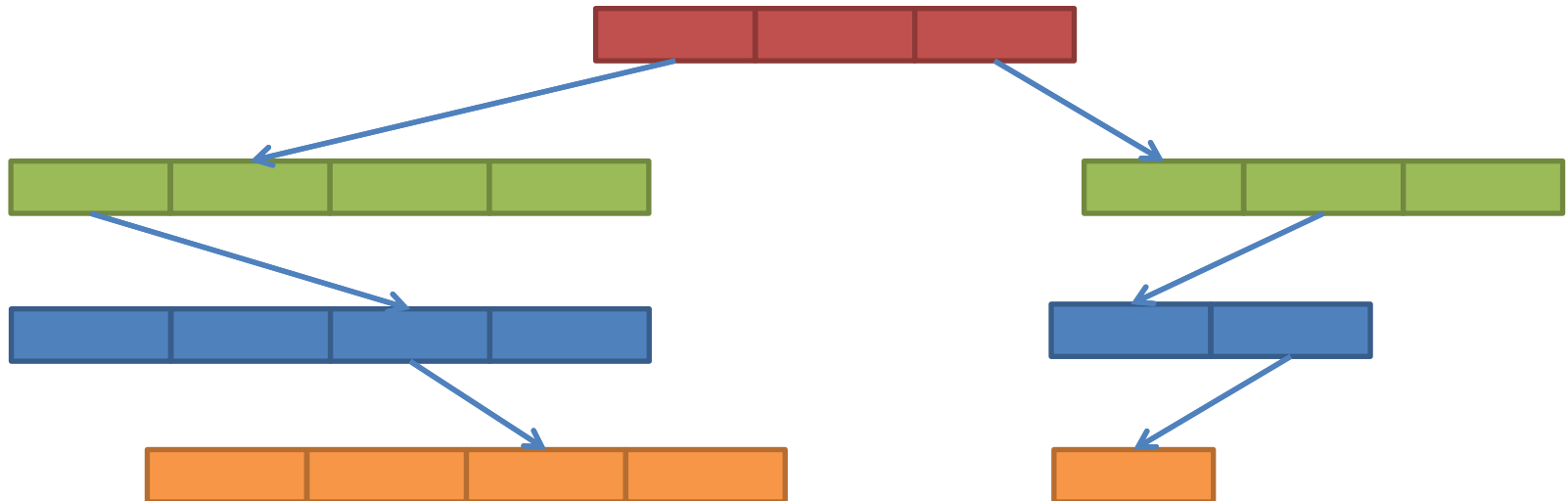


- = Systém souboru počítače může být značně rozsáhlý
 - = množství souborů je třeba je nějak organizovat;
 - = organizace souborů je většinou prováděna ve dvou úrovních
 - = v první úrovni je systém souboru rozdělen na partitions;
 - = uživatel pracuje vždy jen s logickou adresářovou strukturou a logickým systémem souborů
 - = zcela ignoruje problémy fyzické alokace disku souborem;
 - = partitions představují virtuální disky;.
 - = druhá úroveň systému souborů partition obsahuje informace o souborech;
 - = využívá záznamy v adresáři zařízení nebo volume table;
 - = adresář zařízení chová informace jako jméno, lokace, velikost a typ pro všechny soubory na zařízení;

- = Na adresář je možno pohlížet jako na tabulku symbolů
 - = převádí jména souboru na odpovídající záznamy;
- = Operace, které potřebujeme s adresářem provádět:
 - = vyhledání souboru;
 - = vytvoření souboru;
 - = smazání souboru;
 - = výpis adresáře;
 - = přejmenovat soubor;
 - = pohyb v systému souborů;

- = Nejednodušší adresářová struktura je jednoúrovňová
 - = všechny soubory jsou uloženy v jednom adresáři;
 - = vyžaduje minimální podporu;
 - = hlavní nevýhodou adresáře jedné úrovně je zmatek mezi soubory a jejich jmény od více uživatelů;
 - = ve dvouúrovňové adresářové struktuře má každý uživatel svůj vlastní uživatelský adresář souboru - UFD;
 - = každý UFD má stejnou strukturu, ale obsahuje pouze soubory patřící danému uživateli;

- = Více úroňovou struktura
 - = díky generalizaci umožňuje uživatelům vytvářet své vlastní podadresáře a organizovat tak své soubory;
 - = strom je cesta k adresářové struktuře;
 - = strom má kořenový adresář;
 - = ke každému souboru vede jednoznačná cesta od kořenového adresáře přes všechny podadresáře až k cílovému souboru;



- = Adresář obsahuje množinu souborů adresářů nebo podadresářů
 - = adresář je soubor, se kterým OS nakládá speciálním způsobem;
 - = adresáře mají stejný vnitřní formát;
 - = jeden bit v každé položce adresáře identifikuje, zda je daná; položka soubor (0) nebo podadresář (1);
 - = k vytvoření a smazání adresáře užívá systém speciální volání;
- = Při běžné práci má každý uživatel definován aktuální adresář

- = Změna aktuálního adresáře na jiný je prováděna systémovým voláním
 - = jako parametr využívá jméno adresáře a provede predefinování aktuálního adresáře;
 - = k inicializaci aktuálního adresáře uživatele dojde, když systém spustí uživatelskou úlohu, nebo když se uživatel přihlásí do systému;
- = Cesta k souboru může být dvojího druhu:
 - = absolutní (absolute path name);
 - = relativní (relative path name);

- = Jsou-li informace uloženy ve výpočetním systému, je důležité zajistit jejich ochranu před:
 - = fyzickým poškozením (spolehlivost - reliability);
 - = neoprávněným užitím (ochrana – protection);
 - = potřeba ochrany souborů má smysl pouze v případě, že uživatel může přistupovat k souborům jiných uživatelů;
 - = využíváme kontrolovaný přístup;

- = Systém může rozlišovat mezi různými přístupy k souboru:
 - = čtení souboru;
 - = zápis do souboru, nebo jeho přepis;
 - = spuštění, zavedení souboru do paměti a jeho spuštění;
 - = zápis nových informací na konec souboru;
 - = smazání souboru a uvolnění prostoru na disku pro další užití;
 - = výpis jména souboru a dalších informací;
 - = také operace, přejmenování, kopírování nebo editování souboru bývají kontrolovány;

- = Nejčastější způsob ochrany souboru je vytvořit přístup závislý na identifikaci uživatele
 - = nevýhoda tohoto přístupu je velikost seznamu přístupů;
 - = chceme-li umožnit čtení souboru všem uživatelům, musíme do seznamu zaznamenat všechny s právem čtení;
 - = tvorba access listu je pro správce nudná a zdlouhavá;
- = řešením je užití tříd přístupů uživatele k souboru:
 - = vlastník (Owner);
 - = uživatel, který vytvořil soubor se stává jeho vlastníkem;
 - = Skupina (Group);
 - = množina uživatelů, kteří sdílí soubor a potřebují k němu současný přístup se nazývá skupina nebo pracovní skupina;
 - = Ostatní (Universe);
 - = všichni ostatní uživatelé systému;

- = Existují i jiné cesty k problému ochrany souboru
 - = např. zpřístupnit každý soubor heslem
 - = podmínkou je vhodná volba hesla a jeho častá obměna;
 - = jedná se o bezpečnou cestu k zajištění přístupu k souborům jen těm, kteří smějí a tudíž znají heslo;



Univerzita Hradec Králové
Fakulta informatiky a managementu

Děkuji za pozornost...

