

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

Operační paměť je po procesoru druhá nejdůležitější část počítače. Téměř polovina kódu každého procesu pracuje s operační pamětí. V zásadě lze i prohlásit, že čím více má systém operační paměti, tím je rychlejší. Operační paměť byla už součástí Von Neumannovy a Harvardské architektury (u Hv. byla ještě rozdělena na paměť datovou a programovou). Operační paměť (v dnešní době paměť typu RAM) je volatilní, což znamená, že po ztrátě napájení přijdeme o data. Na RAM paměti najdeme čip SPD (Serial Presence Detect), ve kterém je uložena konfigurace RAM. Operační paměť si lze představit jako řadu lineárně uspořádaných buněk očíslovaných pomocí šestnáctkové soustavy (adresy).

Operační paměť má dvě základní sady „drátů“, kterými je připojena k procesoru. Jedna sada se jmenuje adresová sběrnice a slouží k vysílání adresy buňky, ke které se procesor chystá přistoupit. Další sada je datová sběrnice, po které putují mezi procesorem a pamětí data. RAM máme dvojího typu: DRAM a SRAM. SRAM, neboli statická paměť, si své data pamatuje, aniž by se museli obnovovat. DRAM (dynamická RAM) si své data pamatuje tak, že udržují elektrický náboj na své buňce, který se musí periodicky obnovovat. DRAM je tedy sice levnější, ale zato pomalejší.

Cache neboli mezipaměť, je relativně malá (jednotky MB), ale rychlá paměť. Využívá se k nejnужnějším a často používaným datům zařízení. Na rozdíl od bufferu (vyrovnávací paměti) může cache data poskytovat opakovaně. Samotný buffer se používá ke smazání rozdílu v rychlosti dvou zařízení.

Základní rozdíl mezi operační pamětí a úložištěm je ten, že operační paměť v sobě obsahuje pouze aktuálně spuštěné programy a data, zatímco diskové úložiště je trvalé úložiště pro veškerá data daného zařízení.

Registr je ta nejmenší a nejrychlejší paměť v procesoru. Velikost jednoho registru je jeden bit a jejich využití je různé: určitým registrům říkáme kontext, který si ukládáme vždy do program counteru, když provádíme skok v programu. Další využití registrů jsou takzvané příznakové registry, které nám například při provedení instrukce součtu pomohou pomocí flagů zjistit, jestli číslo přeteklo, jestli je 0 nebo jestli je záporné.

Virtualizace paměti

Virtualizace paměti vytváří dvojí iluzi: za prvé tato metoda umožní systému předstírat, že má mnohem více operační paměti, než jí ve skutečnosti doopravdy je. A dále se procesům lze o tom, kde mají data v paměti uložena: dostanou k dispozici adresy, jež neodpovídají skutečnosti.

Principem virtualizace je tedy to, že kód programu pracuje s neexistující, tedy virtuální adresou. Skutečnou, tedy fyzickou adresu, zná pouze operační systém. Děje se tak za prvé z bezpečnostních důvodů a za druhé z důvodu, že v momentě, kdy se proměnné v programu nahradí za adresy není stále ještě možné určit, kam bude program v operační paměti umístěn. Virtualizace umožňuje i zvětšení rozsahu adresního prostoru, což v důsledku zrychluje práci systému. Protože v jednu chvíli pracuje jen tolik procesů, kolik máme jader, tak jen běžící proces musí mít svá data v operační paměti. Ostatní je mohou mít jinde a uvolnit tím prostor. A data právě těchto v aktuální chvíli nepotřebných procesů se ukládají do úložiště (HDD, SSD), jde o tzv. swapování. Další důležitá věc je, že paměť nelze virtualizovat bez podpory procesoru. Překlad z virtuální adresy na fyzickou musí proběhnout extrémně rychle. Lepší procesory (procesory určené pro velké operační systémy) jsou vybaveny obvody, které tohle zvládnou = Memory Management Unit (MMU).

Základní úkoly správy paměti v OS

Jádro OS řídí přidělování paměti procesům i sobě. Udrží informaci o využití/volné paměti (celkově a pro jednotlivé procesy). Zajišťuje ochranu paměti proti vzájemnému přepsání a odpovídá za virtualizaci paměti.

Dynamické přidělování paměti

Paměť je procesům přidělována vždy, když startují. Jádro tedy vybere adresní prostor, jenž je dostatečně velký, aby do něj šlo uložit kód procesu a jeho data. Proces má poté přístup pouze ke své přidělené paměti. V mnoha systémech mohou procesy žádat o další paměť i když už byly spuštěny a jádro jim znovu najde volnou oblast a přidělí ji.

Máme dva druhy přidělování paměti: statické, kdy je dopředu rozhodnuto o tom, jak bude paměť přidělena. Dnes se s takovým přístupem setkáme jen v malých vestavných systémech. Proces má tedy pevně přidělenou oblast, o nic nežádá, nic neuvolňuje. Velké systémy ale musí volit dynamické přidělování, kdy je o rozdělení paměti rozhodováno za běhu systému pomocí výpočtů.

Část operačního systému, která odpovídá za přidělování operační paměti, se anglicky nazývá memory allocator. Při přidělování paměti nevyhnutelně narazíme na jeden problém: po nějakém čase budou v paměti náhodně rozmístěny volné a přidělené bloky různé velikosti. I když zpočátku přidělujeme bloky od začátku paměti, tak jak se jednotlivé procesy ukončují, budou po nich zbývat různé náhodné mezery. Tomuto jevu se říká fragmentace paměti.

K přidělení paměti můžeme přistupovat pomocí čtyř postupů: Za prvé být co nejpresnější. Přidělíme tedy jen tak velký blok paměti, aby přesně odpovídal požadavkům procesu. Tento postup je ale moc pomalý na to, aby se doopravdy používal. Další možnost je přiřadit vždy ten největší možný blok paměti. Zde pořád musíme hledat, byť ne nejlepší možný, ale největší možný. Velké bloky navíc brzy dojdou, takže tento postup také není vhodný. Další postup je vzít první vhodný blok, tedy procházet volné bloky a vzít ten, který je větší než blok, o který žádáme. Poslední přístupem je nejbližší volný. Funguje tedy stejně jako předchozí přístup, ale vždy začíná tam, kde minule skončil s hledáním. Jeho výhoda je, že se fragmenty nenahromadí na začátku datového prostoru, ale jsou rozmístěny rovnoměrně.

Stránkování a další virtualizační metody

Stránkování je technika vizualizace, která umožňuje dobře řídit odkládání části paměti do trvalého úložiště. Logická paměť je rozdělena na úseky pevné délky = stránky. RAM je rozdělena na rámce o velikost stránek, do kterých jsou jednotlivé stránky nahrávány. Pro přístup k datům tedy potřebujeme znát jak číslo stránky, tak informaci o tom, kde se stránka právě nachází – zda je odložena nebo aktivována v konkrétním rámci RAM. Díky MMU si tedy proces myslí, že má data uložena souvisle za sebou, od adresy 0 do maximální přidělené adresy.