

Architektura OS

Jádro OS, typy jádra a jejich vlastnosti

Jádro je základní program operačního systému, jenž pracuje v nechráněném režimu a má plný přístup ke všem zdrojům.

Monolitické jádro: všechny funkce operačního systému jsou implementovány jako funkce jednoho programu, tedy všechny základní funkce OS jsou součástí jádra. Má ho například systém Linux. Jeho velká výhoda je, že je rychlé a pracuje v souvislém paměťovém prostoru (protože jde jen o jeden program). Mezi nevýhody patří to, že se jádro není schopno naučit nic za běhu, proto musí být při sestavení jádra pamatováno na vše, co by mohlo systém během života potkat. V praxi to například znamená, že nelze pracovat se zařízeními typu „plug and play“. Součástí monolitického jádra jsou například i ovladače zařízení. Chceme-li tedy například pracovat s webkamerou, musíme na to myslet už při sestavení jádra. Tato problematika se obchází pomocí takzvaných modulů, tedy doplňkových funkcí, které lze za běhu k jádru přidávat bez nutnosti vypínání jádra. S tím ale přichází dva problémy: fragmentace paměti a bezpečnostní riziko spojené se zaváděním modulů od neproověřených autorů. Moduly totiž pracují v nechráněném režimu. Bezpečnostní hrozby se většinou řeší pomocí certifikace modulů. Závěrem lze ještě říci, že se u monolitických jader nastavuje spousta parametrů, které určují vyhodnocení tisíců maker na základě toho, na jakém zařízení se nacházíme (server/malé vestavné zařízení).

Mikrojádro: jádru necháváme jen to nejnужnější, ostatní funkce jsou odděleny do samostatných programů. Mikrojádro tedy pracuje v nechráněném režimu, programy v režimu chráněném. Mikrojádru tedy zbyde pouze úkony související s procesy: plánovač procesů (scheduler), správa paměti a komunikace mezi procesy. Výhody jsou zřejmé: rozšíření služeb operačního systému nevyžaduje zásah do jádra. První testy mikrojader naznačily, že mohou mít mikrojádra problémy s výkonem kvůli častému přepínání mezi chráněným a nechráněným režimem.

Hybridní jádro: mix obou předchozích typů jader, snaží se využít výhod obou typů. Vznikla tak, že se k mikrojádru postupně přidaly funkce, jež za běhu v nechráněném režimu často způsobovaly přepínání kontextu. Hybridní jádro mají například Windows, kde běží většina systémových služeb v nechráněném režimu. Dalším systémem využívající hybridní jádro je například MacOS a iOS.

Volání systému

Je-li počítač zapnutý, procesor vykonává kód, uložený v operační paměti na adrese, kam odkazuje příslušný registr PC. V jednom časovém okamžiku může procesor vykonat jen jednu instrukci. Má-li procesor více výpočetních jader, vykoná právě tolik instrukcí, kolik má jader. Pokud například aktuálně vykonávaný proces potřebuje otevřít soubor, musí si na pomoc zavolat systém. Tomuto říkáme volání systému. V kódu programu je pak pro otevření souboru nějaká funkce z API operačního systému (nejčastěji nějaká funkce „open“).

Propojení API systému s kódem aplikací

Po zavolání funkce z API je přerušen původní program a jeho kontext (stavové registry) se nahrají do PC (programm counter). Poté je zahájena volaná funkce z API OS. Volání funkce zahrnuje několik kroků: aplikace volá funkci z API s nějakými parametry (cesta k souboru). Funkce z API připraví do registrů odkaz na název souboru a vyvolá softwarové přerušení se správným číslem. To vše probíhá v chráněném režimu. Poté dojde na přepnutí do nechráněného režimu. Proběhne ověření, zda je

systém volán správně. Kontroluje se oprávnění k danému volání a vstupní data. Poté se vykoná kód funkce open. Tedy nalezení souboru, načtení do operační paměti a vytvoření odkazu na začátek souboru (tzv. „handle“). Do registru se poté nahraje odkaz na tento handle, dojde k přepnutí do chráněného režimu a funkce open je ukončena.

Využití chráněného a nechráněného režimu pro práci jádra OS.

Zde jen krátce: jádro vždy běží celé v nechráněném režimu, má tedy přístup ke všem dostupným zdrojům. V nechráněném režimu poběží také funkce API systému, moduly monolitického jádra. Uživatelské programy a tedy jejich procesy poběží v režimu chráněném z bezpečnostních důvodů. Tyto programy se mohou „okusit“ funkce nechráněného režimu právě pomocí volání systémových API funkcí.

Příklady konkrétních architektur (Android, Windows, Linux)

OS Linux využívá jádro monolitické, avšak je možné chování jádra měnit pomocí modulů, které lze přidávat za běhu. Většinou se jedná o řadiče pro zařízení, které může být odpojováno a připojováno.

Android se označuje jako jádro hybridní. Obsahuje ještě softwarovou nadvrstvu nad jádrem, kterou je virtuální stroj, který pouští aplikace v jazyce Java. Veškerá volání systému jdou tedy přes tuto vrstvu.

Windows mají také hybridní jádro. V chráněném režimu pracují subsystémy, mezi které patří například Session Manager, který řídí relace uživatelů. Dalším subsystémem je třeba Local Security Authority Subsystem Service, který je odpovědný za ověřování uživatelů, změny hesel a řízení přístupů.