

Struktura Singularity: Systém je postaven na mikrojádře, které přiděluje zdroje jednotlivým procesům.

applet spadne, shodí s sebou JRE a také všechny ostatní applety, které v JRE běží. V Singularity k ničemu takovému nedojde. Pokud přestane fungovat jeden SIP, zbytku systému se to nijak nedotkne. Singularity tímto navíc chrání i jednotlivé aplikace. Například pokud je ActiveX komponenta, jako součást Internet Exploreru, napadena malwarem a mohla by poškodit systém, bude v Singularity ukončen jen SIP této komponenty. Samotný browser zůstane nedotčen.

Kanály: Efektivní komunikace

Vzhledem k tomu, že SIP každé aplikace běží do značné míry nezávisle, musela se vyřešit komunikace s jádrem. Tu obstarává Application Binary Interface (ABI). Přitom si vystačí jen s velmi málo povely. Pro srovnání: ABI v Singularity přichází s 163 funkcemi, Windows API (Application Programming Interface) obsahuje kolem 14 000 příkazů.

Zapouzdřené SIP komunikují (i mezi sebou) jen prostřednictvím kanálů. Ty vedou přes oblast, kterou kontroluje jádro –

Exchange Heap. Zde se rovněž nacházejí soubory, se kterými procesy právě pracují. Každý soubor na Exchange Heap náleží vždy jen jednomu procesu. Sdílená paměť (společně používaná oblast operační paměti) je pak v Singularity zcela zakázaná, dovoleno je pouze čtení cizí paměti.

Když internetový prohlížeč stáhne soubor z internetu, uloží jej na Exchange Heap. Aby jej mohl uložit na disk, musí prohlížeč otevřít kanál k ovladači souborového systému. Souborový systém pak komunikuje kanálem s ovladačem pevného disku, poté dojde k uložení souboru. Zároveň kanálem předá SIP informace o vlastníkovvi souboru.

Kanály v této podobě se zatím neobjevily v žádném jiném operačním systému.

V blízké budoucnosti však bude velmi obtížné jimi operační systém vybavit. I proto se zřejmě Microsoft brání označování „příští Windows“. Spousta práce totiž čeká i vývojáře aplikací. Současné aplikace, jako je třeba Photoshop nebo Nero, nový systém nebude podporovat, protože jsou napsány v nezabezpečeném jazyce, jako je C nebo C++. Dokonce ani Java nebo .Net aplikace nejsou bez dalších úprav použitelné, protože jim chybí podpora kanálů.

Stabilita: Modrá obrazovka je nemožná

Nyní k jádru operačního systému. V něm se nacházejí ovladače, což jsou nejnepohlednější části systému. 85 procent všech

Nejrychlejší systém

Benchmarky ukazují, kolik taktů procesoru potřebuje systém pro jednotlivé činnosti. Čím nižší hodnota, tím lepší.

ZAVOLÁNÍ JÁDRA-API

Aplikace odešle požadavek na funkci jádra operačního systému

Singularity	80
Windows XP	627
Linux	437
FreeBSD (Unix)	878

POZASTAVENÍ VLÁKNA

Jádro pozastavuje vlákno, aby mohlo začít pracovat jiné vlákno

Singularity	365
Windows XP	753
Linux	906
FreeBSD	911

OBDRŽENÍ/ODPOVĚĎ NA ZPRÁVU

Jeden proces pošle jednobajtovou zprávu jinému procesu

Singularity	1 040
Windows XP	6 340
Linux	5 800
FreeBSD	13 300

VYTVORENÍ/SPUŠTĚNÍ PROCESU

Operační systém vytvoří nový proces a spustí jej

Singularity	388 000
Windows XP	5 380 000
Linux	719 000
FreeBSD	1 030 000

pádů systému zavíní chybný ovladač. A chyby v linuxových ovladačích jsou sedmkrát častější než chyby ve zbytku kódu jádra.

Problémy s ovladači jsou z velké části způsobeny tím, že systém nad nimi nemá kontrolu. Ovladač může spustit kód, přidělit zdroje a způsobit konflikt s jinou systémovou komponentou. Navíc mnoho ovladačů běží s vysokými systémovými právy. Když pak takový ovladač spadne, většinou to znamená pád celého systému.

Singularity má přichystaná dvě opatření, která černé scénáře eliminují. Prvním je Manifest, který musí být součástí každé instalační rutiny ovladače. Manifest není nic jiného než XML soubor, ve kterém je napsáno, jaký typ hardwaru ovladač vyžaduje. Díky této informaci může Singularity již předem rozhodnout, jestli daný ovladač vůbec může s nainstalovaným hardwarem fungovat. Také se tím eliminuje konflikt různých ovladačů. Druhým opatřením je samotné umístění ovladače. Nainstalovaný ovladač již není součástí jádra systému, ale běží jako obyčejná aplikace zabalený v SIP. Pokud ovladač kvůli programové chybě spadne, nemá to žádný dopad na jádro. Singularity jednoduše spustí ovladač znovu a informuje uživatele o problému.

To je výhoda mikrojádra. Co možná nejvíce komponent se spouští až v okamžiku, kdy jsou potřeba. Mikrojádro ovšem není nic nového. Objevilo se již ve výukovém systému Minix v roce 1987. Nasazení v operačním systému se však tehdy nedařilo – až v Singularity. Důvod: Systémy s mikrojádrům jsou příliš pomalé.

Výkon: Odstranění brzd

Všechny x86 procesory mají chráněné režimy, tzv. Ringy. Celkem jsou čtyři, operační systémy však používají jen Ring0 a Ring3. Zatímco procesy s Ring0 mají přístup ke všem částem CPU a operační paměti, procesy s Ring3 nemají žádný přímý přístup k hardwaru.

Ve Windows, Linuxu a Mac OS běží všechny systémově kritické procesy, jako jsou třeba ovladače, v Ring0 neboli v kernel modu. Ostatní aplikace, jako je třeba Firefox, pak běží v user modu. Vlastně je

to tak správně. Je tu však jedno „ale“: Hardwarová ochrana upřednostňuje monolitické jádro, ve kterém jsou všechny ovladače v kernel modu. A právě to činí operační systém zranitelným.

Mikrojádro má jednoduché řešení – ovladače se spustí v Ring3. Systém zůstane bez problémů stabilní, jen se tím sníží výkon: požadavek aplikace v Ring3 na systém musí probulat do Ring0, což stojí čas, jelikož Ring3 nemá přímý přístup ke zdrojům. Kolik času a výkonu se takto ztratí? Ne zrovna málo. Jedno volání přes hranice ringů je pětkrát až desetkrát pomalejší než volání uvnitř ringu.

Proto jsou monolitická jádra, která komunikují jen velmi málo přes ringy, rychlejší než přicházející mikrojádra. Singularity se však chrání softwarově pomocí výše popsaných SIP, takže ringy vůbec nepotřebuje. Jinak řečeno: Singularity běží kompletně v kernel modu, a díky tomu je mnohem rychlejší než všechny ostatní systémy.


Microsoftský tým pracující na Singularity měřil, o kolik je softwarová ochrana běžící v kernel modu rychlejší než použití ringů. Výsledek: Singularity je asi o 30 % rychlejší.

Windows, Linux a další systémy by tak mohly již nyní bez dalších složitých optimalizací běžet až o 30 % rychleji. Musela by se ovšem vyřadit hardwarová ochrana – což by zvýšilo riziko pádu. Singularity se nevzdává vysoké ochrany a přitom se jedná o štíhlý systém.

Singularity využije hardware o 30 % efektivněji

Všechny tyto faktory vedou k excellentním výsledkům, co se týče výkonu (podívejte se na benchmarky vlevo). Aplikace jsou navíc, oproti těm na ostatních systémech, znevýhodněny „bezpečným programováním“.

Singularity se však může v jednotlivých případech vrátit na hardwarovou ochranu. K tomu má systém speciální Sandbox (odkladiště), ve kterém může kontrolovat kód, zda se nejedná o malware. Antivir tak bude skutečně zbytečný.

Dobrou práci odvedli vývojáři Microsoftu. V současnosti je Singularity jen v podobě modrého okna s příkazovým řádkem, experimentům je však tento nový systém zcela otevřený. Rozhodně se bude jednat o nástupce Windows. Bude to zcela nová generace operačního systému. 

AUTOR@CHIP.CZ

Zdroj: Microsoft