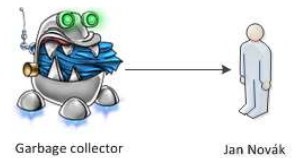


Úkol 01 [1 - 1]

neděle 12. března 2023

13:52



GARBAGE COLLECTOR (GC) a C#

- Pro každý objekt, CLR (common language runtime) je alokována paměť z haldy (heap).
- Jelikož paměť není neomezená je potřeba místo uvolňovat. K tomu slouží právě GC, který se vyhá na haldu kde shromáždí již nepoužívané objekty a uvolní je z paměti.

Paměť

- Každému procesu je přidělen virtuální prostor z fyzické paměti. Každý program pracuje s virtuálním prostorem, nikoliv fyzickou pamětí. Stejně tak GC.
- Ve virtuální paměti existují tzv díry, které správce vyhledává a přiřazuje jednotlivým objektům.
- Virtuální paměť lze dělit na tři druhy
 - Free - volný prostor
 - Reserved - již přidělené
 - Committed - blok předaný do fyzické paměti a není k dispozici k přidělení

Práce GC

- GC pracuje na spravované haldě. Když se GC spustí kontroluje "mrtvé" objekty a objekty, které se nepoužívají.
- Živé objekty zkomprimuje a snaží se uvolnit další místo.
- Halda v paměti je spravována různými generacemi, zpracovává long-lived a short-lived objekty
 - Generace 0 - krátkodobé "dočasné" objekty
 - Generace 1 - tzv. buffer mezi long-lived a short-lived objekty
 - Generace 2 - dlouhodobé objekty, např statické a globální proměnné po určitou dobu běhu.
- Objekty, které nejsou shromážděny v gen 0 jsou přesunuty do gen 1 a pokud ani v gen 1 tak jsou zařazeny do gen 2.

Rozhodování zda je objekt živý

- Udrží všechny vazby(zpracování), které jsou alokována uživatelským kódem nebo CLR
- Sleduje statické objekty když se na ně odkazují jiné objekty
- Používá zásobník používaný stack walkerem a JIT (Just-In-Time compiler)

Spuštění GC

GC se spouští podle následujících podmínek:

- Pokud dojde místo ve virtuálním prostoru.
- Pokud zjistí, že míra živých objektů je vysoká, tak zvýší prahovou hodnotu pro alokaci.
- Je možné zavolat GC.collect(), ale jelikož GC běží paralelně velde aplikace, tedy neustále. Není třeba tuto fci explicitně volat.

(Ne) spravované zdroje

- Spravované - Vše co spadá pod rámec .NET, např. String, int, bool - označení jako spravovatelný kód
- Nespravované - Obecně řečeno knihovny třetích stran na které se odkazuje .NET kód

Čištění nespravovaných zdrojů

- Nespravované objekty je nutné explicitně uvolnit, většinou řešeno systémovým kódem který "obalí" kód. Např. Připojení k DB, stream do souboru, ... GB sleduje životnost jak spravovaných, tak nespravovaných objektů, ale není zodpovědný za uvolňování těch nespravovaných.
- Několik typů jak přistupovat k čištění nespravovaných zdrojů
 - Implementace interface **IDisposable** s metodou dispose
 - Použití bloku **using**, který nakonec po skončení bloky uvolní.

