

# Zálohovací systém

## Klientská aplikace

Implementujte zálohovací aplikaci v jazyce C#. Bude se jednat o nevizuální o konzolovou aplikaci. Klient provádí zálohování dat podle standardizovaného konfiguračního souboru. Zálohovací konfigurace se provádí periodicky v definovaném intervalu.

## Parametry zálohy

**Zdroj** – záloha provádí archivaci více zdrojových složek. Veškeré definované zdrojové složky budou součástí výstupního balíčku zálohy. Zdrojovými cestami jsou vždy složky z místního disku. Příklad:

- C:\Users\Pepa\Dokumenty
- C:\Hry
- D:\Filmy

**Cíl** – záloha definuje více cílových složek. Do těchto umístění se výstupní balíček zrcadlí. Všechna cílová umístění tedy obsahují stejná data. Cílovými cestami jsou vždy složky z místního disku. Příklad:

- D:\Zaloha
- X:\Backup

**Algoritmus** – určuje, jakým algoritmem se bude záloha provádět (full/differential/incremental).

**Retence** – množství historicky uchovávaných záloh. Zálohy starší, než maximální množství uchovávaných záloh budou automaticky odstraněny.

**Časování** – textový řetězec ve formátu CRON.

## Časování

CRON je způsob plánování úkolů v unixových a unixově podobných operačních systémech. Formát CRONu je řetězec, který definuje, kdy a jak často má být určitý úkol spuštěn. Tento řetězec má pět položek oddělených mezerami a vypadá následovně:

\* \* \* \* \*

Každá z těchto pěti položek reprezentuje určitý časový aspekt plánování úkolu. Zde je vysvětlení jednotlivých položek:

1. **Minuta (0-59)**: První položka určuje minutu, kdy se má úkol spustit. Může to být jakákoli číselná hodnota mezi 0 a 59. Například, **5** znamená, že úkol se spustí v páté minutě každé hodiny.
2. **Hodina (0-23)**: Druhá položka určuje hodinu, kdy se má úkol spustit. Může to být jakákoli číselná hodnota mezi 0 a 23. Například, **13** znamená, že úkol se spustí v první hodině odpoledne.
3. **Den v měsíci (1-31)**: Třetí položka určuje den v měsíci, kdy se má úkol spustit. Může to být jakákoli číselná hodnota mezi 1 a 31. Například, **10** znamená, že úkol se spustí desátého dne v měsíci.

4. **Měsíc (1-12):** Čtvrtá položka určuje měsíc, kdy se má úkol spustit. Může to být číselná hodnota od 1 do 12. Například, **6** znamená, že úkol se spustí v červenci.
5. **Den v týdnu (0-6):** Pátá položka určuje den v týdnu, kdy se má úkol spustit. Může to být číselná hodnota od 0 (neděle) do 6 (sobota). Například, **3** znamená, že úkol se spustí ve středu.

Kromě jednotlivých číselných hodnot může symbol **\*** znamenat "každý" a umožňuje tak plánovat úkoly v daném časovém úseku. Například, **\*** ve třetí položce znamená, že úkol bude spouštěn každý den v měsíci.

Příklad CRON řetězce:

- **30 8 \* \* \*:** Úkol se spustí každý den v 8:30 ráno.
- **0 18 \* \* 1-5:** Úkol se spustí každý pracovní den (od pondělí do pátku) v 18:00.
- **0 0 1 \* \*:** Úkol se spustí prvního dne v měsíci v půlnoci.

CRON je velmi flexibilní systém pro plánování úkolů a umožňuje detailní nastavení časování podle potřeb.

## Zálohovací algoritmy

Princip zálohování - Full, Differential a Incremental Backup

### Full Backup (Kompletní záloha):

- Při provádění full backupu jsou zálohována všechna data a soubory z původního zdroje.
- Při prvním spuštění full backupu je vytvořena kompletní kopie všech souborů a dat.
- Každý následující full backup vytváří novou kopii všech souborů, nezávisle na předchozích zálohách.
- Full backup je nejčasově náročnější a zabírá nejvíce místa na úložišti.

### Differential Backup (Diferenční záloha):

- Při provádění differential backupu jsou zálohovány pouze změněné soubory a data od posledního full backupu.
- Na začátku je nutné provést full backup.
- Differential backup ukládá veškeré změny provedené od posledního full backupu, bez ohledu na to, zda byly předchozí změny již zálohovány nebo ne.
- Differential backup je rychlejší než full backup, ale zabírá více místa na úložišti než incremental backup.

### Incremental Backup (Inkrementální záloha):

- Při provádění incremental backupu jsou zálohovány pouze změněné soubory a data od posledního full nebo incremental backupu.
- Na začátku je nutné provést full backup.
- Inkrementální backup ukládá pouze změny od posledního full nebo incremental backupu, což znamená, že ukládá pouze nejnovější změny.

- Inkrementální backup je nejefektivnější z hlediska času a úložného prostoru, protože ukládá pouze nové nebo změněné soubory od poslední zálohy.
- Obnovení dat vyžaduje zpětné aplikování posloupnosti inkrementálních záloh od posledního full backupu.

#### Porovnání:

- Full backup je nejkompaktnější a obsahuje všechna data. Je vhodný pro obnovu kompletního systému, ale vyžaduje nejvíce místa a času pro zálohování.
- Differential backup ukládá všechny změny od posledního full backupu. Jeho obnova je rychlejší než u full backupu, ale zabírá více místa na úložišti.
- Incremental backup ukládá pouze změny od posledního full nebo incremental backupu. Je rychlejší a vyžaduje méně místa na úložišti, ale obnova dat vyžaduje postupné aplikování všech inkrementálních záloh.

Správná volba zálohovacího typu závisí na specifických potřebách a omezeních systému, jako je dostupné místo na úložišti, časové omezení pro zálohování a obnovu dat a rychlost přenosu dat. Je důležité vybrat zálohovací strategii, která nejlépe vyhovuje požadavkům konkrétního prostředí.

## Retence záloh

Retence záloh (anglicky "backup retention") doba, po kterou jsou zálohy uchovávány a dostupné pro obnovu. Retence záloh určuje, jak dlouho budou zálohy uloženy a kdy budou přepsány novými zálohami. Je to důležitý aspekt z hlediska zabezpečení dat a návratnosti do stavu před havárií.

Z hlediska uvedené aplikace budou využívány 2 retenční parametry:

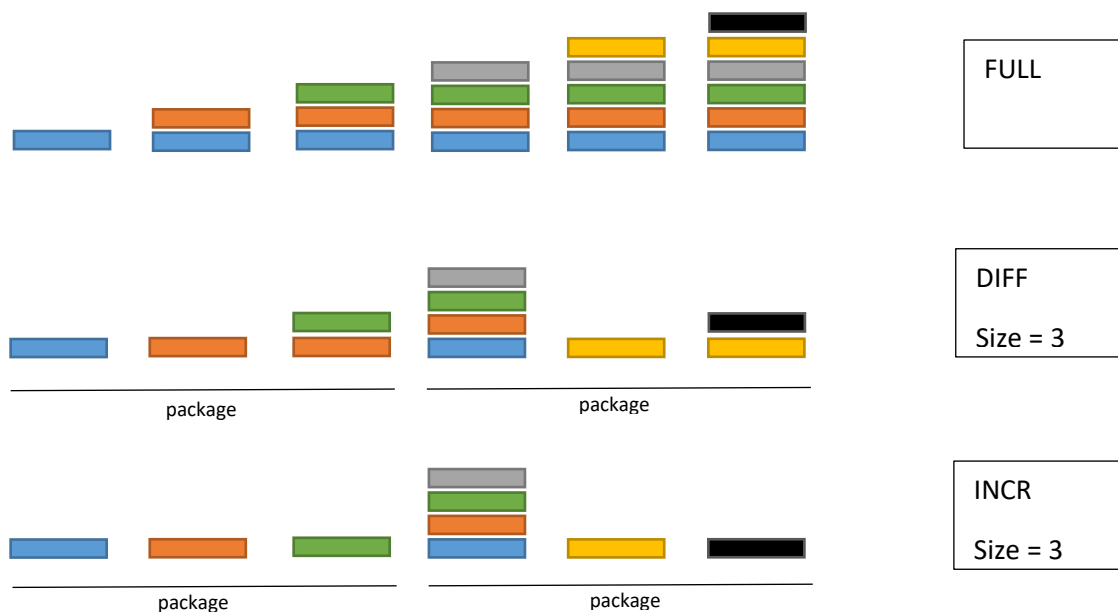
- **Count** – definuje počet uchovávaných balíčků záloh. Zálohy přesahující tento maximální počet záloh, budou odstraněny od nejstarší.
- **Size** – množství částí, z nichž se uvedený balíček zálohy skládá. Význam má při zálohovacích algoritmech Differential Backup a Incremental Backup. V zálohovacím algoritmu Full Backup má vždy pevnou hodnotu 1

### Části zálohy

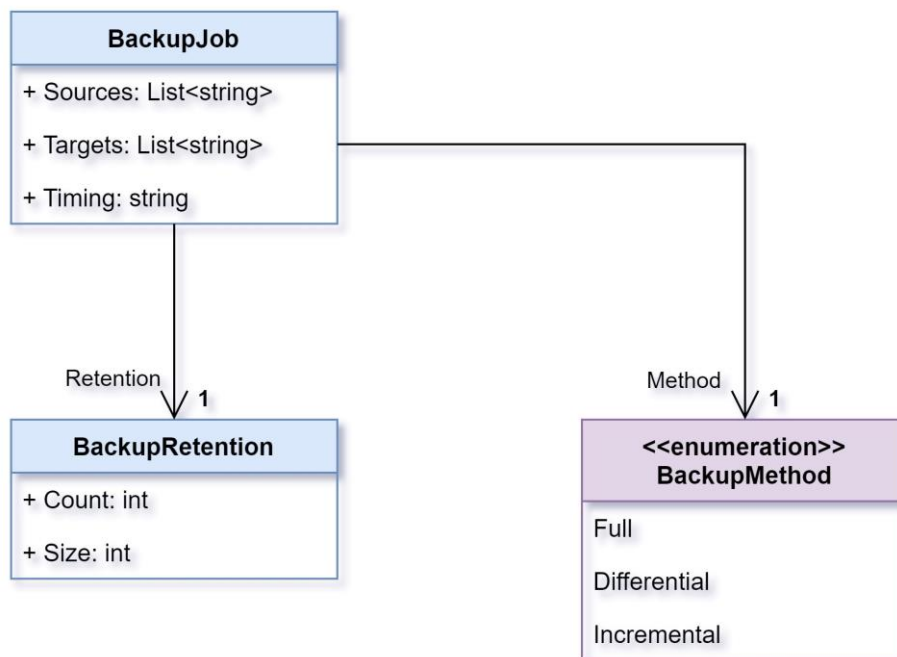
Diferential a Incremental Backup vždy porovnává změny oproti prvnímu/předchozímu stavu. Uvedené části by tedy bez první/všech částí byly neobnovitelné. Tvoří tedy jeden celek (balíček). Hodnota parametru **size** tedy určuje, jaký je maximální počet těchto dílků v rámci jednoho balíčku. Po překročení uvedené hodnoty se začne tvořit nový balíček s novým referenčním bodem. Při odstraňování z důvodu překročení parametru **count** se vždy odstraňuje **celý balíček** (tzn. všechny jeho části). Samotné dílky by byly jinak neobnovitelné.

Full Backup se provádí vždy celá. Nemá tedy žádný referenční bod a hodnota parametru **size** je tedy vždy 1.

## Vizualizace zálohovacích algoritmů



## Datový model



## Formát konfiguračního souboru

Konfigurační soubor je ve formátu JSON. Formát konfiguračního souboru je **závazný**. Povolenými hodnotami parametru method jsou full/differential/incremental.

```
[
  {
    "sources": [
      "C:\\Users\\Pepa\\Dokumenty",
      "C:\\Hry",
      "D:\\Filmy"
    ],
    "targets": [
      "D:\\Zaloha",
      "X:\\Backup"
    ],
    "method": "incremental",
    "timing": "5 4 * * *",
    "retention": {
      "count": 10,
      "size": 5
    }
  },
  ... další objekty BackupJob
]
```