

Plánování úloh v Cron – u

Obchodní akademie, Vyšší odborná škola a Jazyková škola s právem státní
jazykové zkoušky Uherské Hradiště

Autoři: Martin Múčka, Tomáš Šubík, Jan Vyskočil

Datum zpracování: 14. 5. 2024

Úvod

Plánování úloh v systému cron je klíčovou dovedností pro správu serverů a automatizaci úkolů v operačních systémech Unix a Linux. Tento projekt se zaměřuje na vytvoření komplexního průvodce pro využití cron-u v různých praktických scénářích, s cílem usnadnit a zefektivnit každodenní správu systémů pro IT profesionály a administrátory.

Cíl projektu

Cílem tohoto projektu je poskytnout uživatelům podrobný návod, jak efektivně používat cron k plánování úloh. To zahrnuje základní konfiguraci a použití. Projekt bude obsahovat příklady konkrétních skriptů a úloh, které mohou být snadno přizpůsobeny různým potřebám.

Využití v praxi

V praxi je efektivní plánování úloh zásadní pro správu serverů a systémů. Automatizace úloh pomocí Cron může výrazně snížit manuální práci, minimalizovat chyby a zajistit, že důležité úkoly jsou prováděny pravidelně a spolehlivě. Například zálohování dat, aktualizace systémů, čištění logů nebo zasílání pravidelných reportů jsou úkoly, které mohou být automatizovány a tím zvýšit efektivitu a spolehlivost systémové správy.

Jak Cron funguje?

Napišeme příkaz **crontab -e** a nastavíme, kdy chceme, aby se skript spouštěl:

- První hvězdička znamená jakou minutu chceme skript spouštět
- Druhá hvězdička znamená jakou hodinu chceme skript spouštět
- Třetí hvězdička znamená jaký den v měsíci chceme skript spouštět
- Čtvrtá hvězdička znamená jaký měsíc chceme skript spouštět
- Pátá hvězdička znamená jaký den v týdnu chceme skript spouštět

```
# _____ minute (0 - 59)
# _____ hour (0 - 23)
# _____ day of the month (1 - 31)
# _____ month (1 - 12)
# _____ day of the week (0 - 6) (Sunday to Saturday;
#                                     7 is also Sunday on some systems)
#
# * * * * * command to execute
```

Návod na provedení úlohy:

1. Pomocí příkazu **touch** jsme vytvořili soubor **log.txt**
2. Vytvoříme skript, do kterého zapíšeme, aby zapisoval aktuální datum s obsahem domovské složky do souboru **log.txt**
3. Pomocí příkazu **crontab -e** jsme nastavili, aby se skript spouštěl každou minutu
4. Po 2 minutách jsme skontrovali obsah souboru log.txt a bylo tam zapsané aktuální datum a obsah domovské složky
5. Poté jsme nastavili, aby se příkaz spouštěl v pondělí 1 hodinu po půl noci

2.

```
GNU nano 7.2 bash.sh
echo "Aktuální datum $(date)" > ~/log.txt
ls -l ~/ >> ~/log.txt
```

3.

```
* * * * * bash ~/bash.sh
```

4.

```
subik@debian-subik:~$ cat log.txt
Aktuální datum Út 14. května 2024, 20:09:01 CEST
celkem 16
-rw-r--r-- 1 subik subik 595  4. úno 12.21 authorized_keys
-rw-r--r-- 1 subik subik  66 14. kvě 20.03 bash.sh
drwxr-xr-x 2 subik subik 4096 31. říj  2023 directory
-rw-r--r-- 1 subik subik  58 14. kvě 20.09 log.txt
-rw-r--r-- 1 subik subik   0 14. kvě 19.43 log2.txt
subik@debian-subik:~$ _
```

5.

```
0 1 * * 1_bash ~/bash.sh
```



Závěr

Všechny části úlohy se nám povedly, jak napsání skriptu, tak spouštění skriptů v daném čase. Dalším řešením může být například použití: VisualCron, Fcron, Anacron, Systemd, ActiveBatch atd.

Mezi nevýhody cronu mohou patřit následující věci:

- **časově omezené plánování** – cron je založen výhradně na čase, na rozdíl od pokročilejších plánovačů úloh nemůže plánovat úlohy na základě událostí nebo podmínek
- **žádné centralizované řízení / žádná centralizovaná správa** – V organizacích s více servery neexistuje žádný integrovaný způsob, jak centrálně spravovat nebo monitorovat úlohy cron ve všech počítačích.
- **potencionální vznik tichých poruch** – Nemusíme si všimnout neúspěšných úloh

Mezi výhody patří:

- jednoduchost
- dostupnost
- plánování podle potřeb uživatele

Rozdělení práce:

Tomáš Šubík – psaní kódu

Martin Múčka – vyhledávání informací

Jan Vyskočil – psaní dokumentace

Zdroje:

<https://research.aimultiple.com/cron-alternative/>

<https://chatgpt.com/>