

Operační systémy

Spouštění úloh v daných časech
Laboratorní úloha

Mgr. Tomáš Hudec



Spouštění úloh v daných časech (laboratorní cvičení)

▼ Obsah

- 1 Cíle a předpoklady
 - 1.1 Použité konvence
 - 2 Možnosti plánování úloh uživatelem
 - 3 Jednorázové úlohy – at
 - 3.1 Příklady
 - 3.2 Výstup úloh
 - 4 Periodické úlohy – crontab
 - 4.1 Příklady
 - 4.2 Nepravidelný běh systému – anacron
 - Použité zdroje a další literatura
-

1 Cíle a předpoklady

Tento text je laboratorním cvičením a předpokládá se, že podle uvedeného postupu bude čtenář/student příklady sám provádět. Cílem je:

- seznámení se s možnostmi plánování úloh na daný čas pomocí příkazů `at` a `crontab`,
- procvičení příkazu `at`,
- vytvoření tabulky pro `crontab`,
- přečtení výstupů příkazů naplánovaných a provedených neinteraktivně.

Předpokládá se přístup k posixovému systému (typicky nějaká distribuce GNU/Linux) s nainstalovanými příslušnými službami (`atd`, `cron`, případně též `anacron`). Ty nemusí být automaticky předinstalovány; na distribuci Debian GNU/Linux a odvozených je lze doinstalovat příkazem:

```
sudo apt install at cron anacron
```

Nepoužívá-li se na systému příkaz `sudo`, je třeba příkaz provést pod uživatelem `root` bez `sudo`. Na uživatele `root` se v tomto případě dá přepnout příkazem `su` a zadáním administrátora hesla. Pro ostatní distribuce GNU/Linuxu a posixové systémy nepoužívající `apt` konzultujte příslušnou dokumentaci pro instalaci softwaru.

1.1 Použité konvence

Jména souborů a příkazů jsou vyznačována takto: *soubor, příkaz*

```
příkaz # komentář
```

Obsahuje-li příkaz nebo uživatelský vstup proměnnou (část, kterou je třeba nahradit skutečnými údaji), je použito následujícího vyznačení:

```
příkaz doslovný-argument proměnná argument-s-proměnnou
```

V ukázkových výpisech z terminálu je použito následujícího vyznačování:

```
výzva-shellu$ uživatelský vstup (příkaz, proměnná) stisknuté klávesy nebo komentář pro vstup  
výstup příkazu
```

2 Možnosti plánování úloh uživatelem

V Unixu a odvozených systémech včetně GNU/Linux jsou k dispozici následující příkazy pro plánování úloh:

- `at` pro naplánování jedné úlohy na konkrétní čas,
- `crontab` pro manipulaci s tabulkou opakovaně prováděných úloh.

O provádění naplánovaných úloh se starají démony `atd` a `cron`, případně `anacron`.

Démon `atd` je zodpovědný za provádění jednorázových úloh zadaných příkazem `at`. Používání příkazu `at` lze omezit pomocí souborů `/etc/at.allow` a `/etc/at.deny`, do kterých lze vložit jména uživatelů systému (jedno jméno na řádek). Existuje-li `/etc/at.allow`, pouze uživatelé v něm uvedení smíjí používat příkaz `at`. Neexistuje-li `/etc/at.allow`, všichni uživatelé vyjma těch vedených v `/etc/at.deny` smíjí používat příkaz `at`. Prázdný soubor `/etc/at.deny` (typická konfigurace) znamená, že všichni uživatelé smíjí používat `at`. Neexistuje-li ani jeden z uvedených souborů, smí používat `at` pouze administrátor (uživatel `root`).

Opakovaně prováděné úlohy mají na starost démony `cron`, případně `anacron`. Řídí se příslušnými tabulkami, které specifikují časy a periody pro úlohy. Možnost periodického plánování úloh uživateli lze omezit pomocí souborů `/etc/cron.allow` a `/etc/cron.deny`. Existuje-li `/etc/cron.allow`, pouze uživatelé v něm uvedení smíjí používat příkaz `crontab`. Neexistuje-li `/etc/cron.allow`, všichni uživatelé vyjma těch vedených v `/etc/cron.deny` smíjí používat příkaz `crontab`. Neexistuje-li ani jeden z uvedených souborů, záleží na systémové konfiguraci, jestli smí používat `crontab` pouze administrátor (uživatel `root`), nebo všichni.

Nápovědu pro jednotlivé příkazy a formáty souborů lze kromě internetu (viz [Použití zdroje](#)) získat v manuálových stránkách přímo na systému:

```
man at           # příkaz at
man crontab      # příkaz crontab
man 5 crontab    # konfigurační soubor pro crontab
```

3 Jednorázové úlohy – at

Chce-li uživatel spustit příkaz v konkrétní dobu místo v okamžiku zadání, lze využít příkaz `at` *čas*. Čas se obvykle udává ve tvaru `H:MM` a bez uvedení konkrétního data znamená „během následujících 24 hodin“, (tj. zadáním času, který ten den již byl, znamená provedení úlohy zítra v daný čas). Nepoužívá-li se 24hodinový formát, lze k času připojit (bez mezery) `AM` (dopoledne) nebo `PM` (odpoledne). Předdefinované jsou též slovní výrazy `now`, `midnight`, `noon` a `teatime` (tj. 16:00).

Zadání lze upřesnit připojením data za čas. Podporováno je několik formátů: `měsíc den [rok]` (slovně anglicky, `rok` lze vynechat) `MMDD[CC]YY, MM/DD/[CC]YY, DD.MM.[CC]YY` nebo `[CC]YY-MM-DD`. Lze využít též slovní výrazy `today` a `tomorrow`.

Za časový údaj lze též připojit časový interval ve formátu: `+ počet jednotka`, kde `počet` je číslo a `jednotka` je `minutes`, `hours`, `days` nebo `weeks`, přičemž slova lze zkrátit a na jednotné/množné číslo se nebere ohled. Např. uvedení `now + 10 min` znamená naplánování úlohy za 10 minut.

Příkazy pro úlohu čte příkaz `at` ze standardního vstupu nebo ze souboru zadaného jako argument přepínače `-f`.

Pro zobrazení seznamu naplánovaných úloh se používá příkaz `atq` (nebo `at -q`). Příkaz vypisuje pouze čísla úloh, časy plánovaného spuštění, označení fronty a vlastníky úloh. Obsah konkrétní úlohy daného *čísla* lze získat příkazem `at -c číslo`. Zobrazí se nejen zadané příkazy, ale celý skript, který obsahuje nastavení z příkazového řádku v době zadávání úlohy: zachovává se pracovní adresář, proměnné prostředí, a maska oprávnění (`umask`), která ovlivňuje práva nově vytvářených souborů a adresářů.

3.2 Příklady

Vyzkoušejte si příkazy (Ctrl + D ukončuje vstup):

```
user@host:~$ date
po 20. května 2024, 18:23:13 CEST
user@host:~$ at now + 5 min
warning: commands will be executed using /bin/sh
at> date
at> CTRL+D
job 157 at Mon May 20 18:28:00 2024
user@host:~$ echo 'date' | at 13:13
warning: commands will be executed using /bin/sh
job 158 at Tue May 21 13:13:00 2024
user@host:~$ atq
157      Mon May 20 18:28:00 2024 a user
158      Tue May 21 13:13:00 2024 a user
user@host:~$ at -c 157
#!/bin/sh
# atrun uid=1000 gid=1000
# mail user 0
umask 22
PWD=/home/user; export PWD
LOGNAME=user; export LOGNAME
HOME=/home/user; export HOME
LANG=cs_CZ.UTF-8; export LANG
USER=user; export USER
PATH=/usr/local/bin:/usr/bin:/bin; export PATH
cd /home/user || {
    echo 'Execution directory inaccessible' >&2
    exit 1
}
date

user@host:~$ at -c 158 | sed '1,/^\}$d' # sed odstraní řádky od prvního do řádku s pouhou }
date

user@host:~$ atrm 157
user@host:~$ atq
158      Tue May 21 13:13:00 2024 a user
user@host:~$ atrm 158
user@host:~$ atq
user@host:~$
```

3.3 Výstup úloh

Pokud úloha vygeneruje jakýkoliv (standardní či chybový) výstup, bude zaslán uživateli jako e-mailová zpráva. Není-li systém nakonfigurovaný jinak, znamená to doručení do lokální schránky (v `/var/spool/mail`). Tělo zprávy je tvořeno výstupem úlohy, předmět je „Output from your job *čísla*“. Pokud úloha žádný výstup nevygeneruje, žádný email se neposílá. Změnit zaslání zprávy lze uvedením přepínačů příkazu `at` na příkazovém řádku:

- `-m` – zasílá zprávu, i když úloha nevygeneruje výstup,
- `-M` – nezasílá zprávu nikdy.

V terminále si lze přečíst e-mail pomocí některého terminálového poštovního klienta (viz [7]), lze doporučit např. `alpine`; nainstalovat lze na Debianu a spol. pomocí:

```
sudo apt install alpine
```

Po spuštění `alpine` je doporučeno nastavit zobrazení v `UTF-8` pro zprávy bez specifikace kódování: V hlavním menu postupně vybrat `S` Setup → `C` Config → nalistovat nebo (vyhledat pomocí `W` WhereIs) „Unknown Character Set“, po stisknutí `Enter` zadat „UTF-8“ a potvrdit (`Enter`). Opustit nastavení: `E` Exit Setup a potvrdit uložení změn (`Enter`).

Ovládání programu `alpine` je celkem intuitivní, nápověda pro klávesové zkratky je ve spodních dvou řádcích terminálového okna. Je-li možných příkazů více, než se vejde na obrazovku, klávesa `O` (Other cmds) cyklicky zobrazuje další možnosti. Klávesou `I` (Index) nebo pomocí kurzorových kláves se lze dostat na seznam zpráv ve schránce, opustit `alpine` lze klávesovou zkratkou `Q` (Quit).

K vyzkoušení:

```
user@host:~$ at now <<< "date"
warning: commands will be executed using /bin/sh
job 159 at Mon May 20 18:35:00 2024
user@host:~$ alpine
```

4 Periodické úlohy – crontab

Provádění periodických úloh se řídí tabulkou, která se nastavuje příkazem `crontab`. Základní možnosti jsou:

- `-l` – zobrazit stávající tabulku,
- `-e` – spustit editaci tabulky,
- `-r` – odstranit tabulku,
- `soubor` – přepsat tabulku daným *souborem*.

Pro editaci se spustí výchozí systémový editor. Distribuce Debian dává uživateli při prvním spuštění systémového editoru na výběr z instalovaných editorů. Pro začátečníky lze doporučit `nano` nebo `mcedit`, zkušení uživatelé mohou sáhnout po `vim` (Vi IMproved, viz [8]). Pro spuštění konkrétního editoru lze použít také systémovou proměnnou `EDITOR`. Je-li tato proměnná nastavená, spustí se pro editaci příkaz v ní uvedený. Příklad nastavení v shellu bash:

```
export EDITOR=nano
```

Pro jednorázové nastavení editoru pouze pro příkaz `crontab` (nemění obsah proměnné pro ostatní příkazy) lze použít:

```
EDITOR=nano crontab -e
```

Formát tabulky je popsán v manuálové stránce `crontab(5)` [4]. Prázdné řádky, úvodní prázdné znaky na řádku a komentáře (řádky začínající znakem `#`) jsou ignorovány. Proměnné prostředí lze nastavovat pomocí:

```
NAME=value
```

Např.:

```
SHELL=/bin/bash
```

Vlastní tabulka pro úlohy má sloupce oddělené mezerami nebo tabulátory, řádky jsou ukončené novým řádkem (LF, `\n`). Jednotlivé sloupce určují časy provádění úlohy, příkaz (s případnými argumenty) je uveden na řádku poslední. Systémová tabulka (`/etc/crontab`) má před příkazem ještě sloupec navíc: jméno uživatele (login). Jednotlivé časové sloupce jsou po řadě:

- *minuta* (0–59),
- *hodina* (0–23),
- *den měsíce* (1–31),
- *měsíc* (1–12 nebo třípísmenná anglická zkratka),
- *den týdne* (0–7 nebo třípísmenná anglická zkratka, 0 = 7 = neděle, 1 = pondělí, ...).

Časových hodnot smí být ve sloupci více: oddělují se čárkou (bez mezer). Více hodnot po sobě lze zkrátit rozsahem *od-do*. Seznamy a rozsahy lze kombinovat. Místo celého rozsahu hodnot *první-poslední* lze použít znak `*`. Příklady: `1, 5, 8` nebo `2, 4, 6, 12-16, 20`

Rozsahy s krokem větším než jedna lze zapsat ve tvaru *rozsah/krok*. Např. zápis `0-23/4` nebo `*/4` ve sloupci hodin znamená každou čtvrtou hodinu počínaje první hodnotou z rozsahu (zde `0`), tj. `0, 4, 8, 12, 16, 20`.

Zadání úlohy: Na základě uvedených informací zapište zkráceně rozsah obsahující každou lichou hodinu.

Daný příkaz bude prováděn ve všech časech, které vyhovují kombinaci specifikovaných hodnot. Pokud je hodnota omezená (různá od `*`) v obou sloupcích pro den (v měsíci i v týdnu), provede se příkaz, když vyhovuje kterákoliv hodnota. Příklad: `15 5 8, 12 * 3 příkaz` znamená provádění příkazu vždy 8. a 12. dne v (každém) měsíci a také každou středu.

4.4 Příklady

Vyzkoušejte si vytvoření tabulky pro `cron`. Po vyzkoušení tabulku odstraňte nebo všechny (nepotřebné) úlohy zakomentujte, aby se vám neplnila schránka chybovými zprávami o neexistujících příkazech.

```
user@host:~$ crontab -l
no crontab for user
user@host:~$ crontab -e
no crontab for user - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /usr/bin/vim.gtk3
 2. /bin/nano          <---- easiest
 3. /usr/bin/vim.basic
 4. /usr/bin/mcedit
 5. /usr/bin/vim.tiny

Choose 1-5 [2]: 2
spuštění editoru, editace souboru
# Příkazy budou spuštěny v shellu bash:
SHELL=/bin/bash
# Žádný e-mail s výstupem úloh se nebude posílat:
MAILTO=""
# Uvedené příkazy jsou pouze ilustrativní, nejedná se o skutečné programy.
# min  h   den   měsíc  den t.  příkaz
18     4     *     *     6     weekly-backup # každou sobotu v 4:18
11     5     1     *     *     monthly-backup # vždy 1. v měsíci v 5:11
54     7,19  */5   *     *     check-status  # v 7:54 a v 19:54, když je datum dělitelné 5
42     21    24    5     *     mail -s "It's Towel Day tomorrow!" $LOGNAME %Don't forget to
prepare a towel!
42     8     25    5     *     mail -s "It's Towel Day today!" $LOGNAME %Don't forget to
take a towel!
# První znak % znamená, že následuje vstup pro uvedený příkaz.
# Každý další znak % se na vstupu nahradí novým řádkem.
# Takto lze zapsat víceřádkový vstup, aniž by bylo nutné jej vkládat do externího souboru.
uložení změn, ukončení editoru
crontab: installing new crontab
user@host:~$ crontab -r # odstranění tabulky
user@host:~$ crontab -l
no crontab for user
```

Další příklady jsou uvedené v manuálové stránce `crontab(5)` [4].

Zadání úloh:

1. Na základě uvedených informací a prostudování manuálu pro konfiguraci tabulky `crontab` [4] se pokuste určit nejkratší a nejdélší možný interval spuštění, který je možné nakonfigurovat.
2. Zapište konfiguraci tabulky pro příkaz prováděný vždy 2. úterý v měsíci.

4.5 Nepravidelný běh systému – anacron

Démon `cron` předpokládá běh systému 24/7. To však pro desktopy obvykle neplatí. Úlohy vycházející na dobu, kdy systém neběží, `cron` neprovádí. **Zadání úlohy:** Zkuste si zdůvodnit, proč.

Aby se periodicky plánované úlohy prováděly i na systémech neběžících trvale, je třeba nainstalovat démon `anacron`. Ten je nastaven tak, aby se úlohy prováděné v denních, týdenních a měsíčních intervalech provedly i mimo standardní čas, byli systém po příslušnou dobu offline.

Na systému Debian GNU/Linux jsou pro snadnou konfiguraci připraveny adresáře, do kterých lze vložit skripty, které se mají provádět v intervalech hodina, den, týden a měsíc. Démon `cron` je nastaven, aby v případě existence démonu `anacron` prováděl pouze úlohy v hodinových intervalech. O vykonání úloh denních, týdenních a měsíčních se stará `anacron`. Konfigurace v souboru `/etc/crontab` vypadá následovně:

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab' command
# to install the new version when you edit this file and files in /etc/cron.d.
# These files also have username fields, that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
#.----- minute (0-59)
#| .----- hour (0-23)
#| | .----- day of month (1-31)
#| | | .----- month (1-12) OR Jan, Feb, Mar, Apr, ...
#| | | | .-- day of week (0-6) (Sunday=0 or 7) OR Sun, Mon, Tue, Wed, Thu, Fri, Sat
#| | | | | user-name  command to be executed
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * * root    test -x /usr/sbin/anacron || { cd / && run-parts --report
/etc/cron.daily; }
47 6 * * * 7 root    test -x /usr/sbin/anacron || { cd / && run-parts --report
/etc/cron.weekly; }
52 6 1 * * * root    test -x /usr/sbin/anacron || { cd / && run-parts --report
/etc/cron.monthly; }
```

Je-li třeba spouštět úlohu např. v týdenním intervalu, stačí skript pro její provedení umístit do adresáře `/etc/cron.weekly`. Pokud systém v neděli v 6:47, kdy jsou týdenní úlohy plánované, nepoběží a démon `anacron` není nainstalován, úlohy se neprovedou. V případě přítomnosti démonu `anacron` budou úlohy prováděné právě démonem `anacron` ve zhruba týdenním odstupu, případně později, pokud systém delší dobu nebyl online. Příslušná konfigurace je v souboru `/etc/anacrontab`. Viz též [6].

5 Shrnutí

Naplánování jednorázové úlohy na daný čas:

```
at 15:27 <<< "příkaz1"          # v 15:27 během následujících 24 hodin
echo "příkaz2" | at 21:35 tomorrow # ve 21:35 následující den
echo "příkaz3" | at 12:40 + 1 week # za týden ve 12:40
```

Tabulka `crontab` pro `cron`, opakované provádění úloh:

```
crontab -l # zobrazí stávající tabulku
crontab -r # odstraní stávající tabulku
crontab -e # spustí editor pro editaci tabulky
```

Příklad konfigurace pro `crontab` – obsah souboru:

```
SHELL=/bin/bash
MAILTO=""
# min h den měsíc den t. příkaz
*/20 * * * * check-disk-space # každých 20 minut
13 */2 * * * win-nuke-00B-p139 # každé 2 hodiny
42 4 * * * auto-update-packages # každý den ve 4:42
15 17 13 * * check-auto-payment # v 17:15 vždy 13. v měsíci
0 14 6 2 * mail -s "Anniversary reminder" $LOGNAME %Don't forget to buy
flowers!
# každý rok 6. února ve 14:00 zaslat uživateli (lokální) e-mail s daným předmětem a obsahem
```

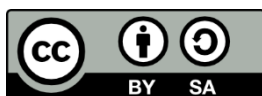
Použité zdroje a další literatura

- [1] `at(1)` [online]. 2009-11-14 [cit. 2024-05-20]. Manuálová stránka. URL: <https://manpages.debian.org/at.cs>
- [2] Přeložil KAVKA, Ondřej: `atd(8)` [online]. 2009-11-14 [cit. 2024-05-20]. Manuálová stránka. URL: <https://manpages.debian.org/atd.cs>
- [3] STEPANOV, Konstantin – DETISTE, Alexandre: `crontab(1)` [online]. 2014-12-10 [cit. 2024-05-20]. Manuálová stránka. URL: <https://manpages.debian.org/systemd-cron/crontab.cs>

- [4] VIXIE, Paul – GREENLAND, Steve – FERNANDEZ-SANGUINO, Javier – KASTNER, Christian: *crontab(5)* [online]. 2014-07-03 [cit. 2024-05-20]. Manuálová stránka. URL: <https://manpages.debian.org/systemd-cron/crontab.5.cs>
- [5] VIXIE, Paul – GREENLAND, Steve – FERNANDEZ-SANGUINO, Javier – KASTNER, Christian: *cron(1)* [online]. 2010-04-19 [cit. 2024-05-20]. Manuálová stránka. URL: <https://manpages.debian.org/cron.cs>
- [6] TZUR, Itai: *anacron(1)* [online]. 2018-11-30 [cit. 2024-05-20]. Manuálová stránka. URL: <https://manpages.debian.org/anacron.cs>
- [7] SAIVE, Ravi: *8 Best Command-Line/Terminal Email Clients for Linux* [online]. TecMint, 2023-11-01 [cit. 2024-05-20]. URL: <https://www.tecmint.com/best-commandline-email-clients-for-linux/>
- [8] SATRAPA, Pavel: *Vim: skutečný editor textů* [online]. Technická univerzita v Liberci, NTI, © 1997–2016 [cit. 2024-05-20]. URL: <https://www.nti.tul.cz/~satrapa/docs/vim/>

Vytvořeno v rámci projektu **Digitalizace studijních Agend, Nové Technologič, systémy a přístupy k výuce na UPCE**, reg. č. NPO_UPCE_MSMT-16591/2022.

Toto dílo podléhá licenci Creative Commons BY 4.0. Pro zobrazení licenčních podmínek navštivte <https://creativecommons.org/licenses/by-sa/4.0/>.



Financováno
Evropskou unií
NextGenerationEU



Národní
plán
obnovy

MSMT
MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY