

# Procesy a úlohy

PROCESY	
<b>chkconfig</b> [<volba> [<sluzba> [<dostupnost>]]]	<b>--add</b> <sluzba> přidá službu pod správu chkconfig, <b>--del</b> <sluzba> odstraní službu, <b>--level</b> <uroven_behu_systemu> určí úroveň běhu systému, <b>--list</b> / <b>--list</b> <sluzba> vypíše všechny služby / danou službu a jejich/její dostupnost {off   on} na všech úrovních běhu systému <pre># chkconfig httpd on</pre> (povolí službu na úrovních běhu systému 2, 3, 4 a 5) <pre># chkconfig --level 234 qemu off</pre> (zakáže službu na úrovních běhu systému 2, 3 a 4) <pre>\$ chkconfig --list   egrep -i "rexe rlogin rsh"</pre> (vypíše dané služby a jejich dostupnost na všech úrovních běhu systému)
<b>systemctl</b> <prikaz> [<sluzba ...>] (platí od RHEL 7)	<b>enable</b> povolí službu, <b>disable</b> zakáže službu, <b>is-enabled</b> zjistí dostupnost služby („service unit“) automaticky po startu systému, <b>mask</b> zabrání automatickému či manuálnímu spuštění služby, <b>unmask</b> zruší účinek příkazu „mask“, <b>--user</b> spravuje uživatelské systemd soubory (služby) <pre># systemctl enable httpd</pre> (povolí službu automaticky po startu systému) <pre># systemctl mask postfix.service</pre> (zabrání automatickému či manuálnímu spuštění služby) <pre>\$ systemctl list-unit-files -t service</pre> (vypíše všechny služby a jejich dostupnost)
<b>service</b> [<sluzba>] <prikaz>	<b>start</b> spustí službu, <b>stop</b> zastaví službu, <b>restart</b> restartuje službu, <b>reload</b> znovu načte konfigurační soubor služby, <b>status</b> zobrazí stav služby <pre># service cups restart  # /etc/init.d/cups restart</pre> (restartuje danou službu) <pre># service --status-all</pre> (zobrazí stav všech služeb)
<b>systemctl</b> <prikaz> [<sluzba ...>] (platí od RHEL 7)	<b>start</b> spustí službu, <b>stop</b> zastaví službu, <b>restart</b> restartuje službu, <b>reload</b> znovu načte konfigurační soubor služby, <b>status</b> zobrazí stav služby („service unit“), <b>is-active</b> vypíše, zda je služba aktivní (běží) nebo neaktivní, <b>--user</b> spravuje uživatelské systemd soubory (služby) <pre># systemctl restart cups</pre> (restartuje danou službu) <pre># systemctl list-units -t service</pre> (zobrazí stav nainstalovaných služeb) <pre># systemctl list-units -t service --state=running</pre> (zobrazí běžící služby) <pre># systemctl list-units -t service --all</pre> (zobrazí stav všech služeb)
<b>systemctl list-jobs</b> (platí od RHEL 7)	vypíše všechny úlohy, které systemd aktuálně provádí

PROCESY	
<prikaz> [<argument ...>]	spustí spustitelný program (binární soubor) v aktuálním shellu; program je vestavěný příkaz shellu či příkaz nalezený v \$PATH nebo v zadané cestě \$ date
<b>bash</b>   <b>sh</b> [<prikaz   soubor> [<argument ...>]]	spustí shell, příkaz či soubor (i bez nastavených práv ke spuštění) ve vnořeném shellu; soubor hledá v pracovním adresáři a v \$PATH, <b>-r</b> spustí omezený shell, <b>-c</b> <retezec> čte příkazy z daného řetězce, <b>-s</b> čte příkazy ze STDIN, <b>-x</b> spustí skript v debug režimu \$ sudo bash -c "echo 'Vítejte na moji strance' > /var/www/html/index.html" (přesměruje výstup příkazu do daného souboru bez nutnosti přihlásit se na účet uživatele root)
<b>source</b>   <b>.</b> <soubor> [<argument ...>]	spustí daný soubor (i bez nastavených práv ke spuštění) v aktuálním shellu; soubor hledá v pracovním adresáři a v \$PATH \$ . ~/.bash_profile (obnoví profil uživatele bez nutnosti opětovného přihlášení)
<soubor> [<argument ...>]	spustí spustitelný soubor ve vnořeném shellu; soubor hledá v \$PATH nebo v zadané cestě \$ ./skript
<b>command</b> <prikaz> [<argument ...>]	spustí vestavěný příkaz shellu nebo příkaz nalezený v \$PATH, přičemž ignoruje jakoukoliv funkci shellu pojmenovanou shodně s názvem příkazu (shell hledá název příkazu v tomto pořadí: funkce shellu, vestavěné příkazy shellu, spustitelné soubory v \$PATH)
<b>builtin</b> <prikaz> [<argument ...>]	spustí vestavěný příkaz shellu, přičemž ignoruje jakoukoliv funkci shellu pojmenovanou shodně s názvem příkazu (shell hledá název příkazu v tomto pořadí: funkce shellu, vestavěné příkazy shellu, spustitelné soubory v \$PATH)
<b>exec</b> [<prikaz> [<argument ...>]] [<presmerovani ...>]	nahradí shell daným příkazem (nevytvoří se žádný nový proces) nebo provede přesměrování \$ exec bash (nahradí shell bashem) \$ exec > output.txt (přesměruje veškerý výstup do souboru „output.txt“ pro aktuální proces shellu)
<b>pstree</b> [<PID   uzivatel>]	zobrazí stromovou strukturu procesů v abecedním pořadí, počínaje procesem "init" (systemd) nebo daným PID či procesy vlastněnými daným uživatelem, <b>-n</b> v číselném pořadí (podle PID), <b>-p</b> vypíše PID, <b>-u</b> vypíše uživatele

PROCESY	
<p><b>ps</b></p>	<p>zobrazí statický seznam běžících procesů přihlášeného uživatele na daném terminálu (PID - identifikační číslo procesu, TTY - terminál, ze kterého byl proces spuštěn (znak „?“ znamená, že s procesem není spojen žádný terminál), TIME - čas, po který byl proces zpracováván procesorem, CMD - příkaz či proces a jeho parametry), <b>-e</b> všechny procesy všech uživatelů, <b>-f</b> podrobný výpis (navíc zobrazí UID - vlastníka procesu, PPID - rodičovský proces daného procesu, C - využití procesoru, STIME - čas spuštění procesu), <b>-l</b> úplný výpis (navíc zobrazí F - příznak procesu (0 - běžný proces, 1 - rozvětvený, ale nespuštěný proces, 4 - proces běžící s právy roota, 5 - kombinace předchozích hodnot), S - stav procesu („R“ běžící proces, „S“ spící proces (čekající na dokončení události), „D“ spící proces nereagující na signály (obvykle I/O), „T“ pozastavený proces, „Z“ zombie (defunct) proces - ukončený proces, který stále zabírá PID, jelikož rodičovský proces si není vědom jeho stavu ukončení), C - vytížení procesoru v %, PRI - priority procesu, NI - hodnotu priority zadanou příkazem „nice“, SZ - celkovou velikost paměti v blocích, kterou proces zabírá, WCHAN - adresu funkce jádra, kde proces čeká na určitou událost (u běžícího procesu je „-“)), <b>-p</b> &lt;PID&gt; daný proces, <b>-C</b> &lt;program&gt; procesy daného programu, <b>-U</b> &lt;uživatel   UID&gt; procesy daného uživatele, <b>-G</b> &lt;skupina   GID&gt; procesy dané skupiny, <b>-t</b> &lt;terminal&gt; procesy na daném terminálu, <b>-o</b> &lt;format&gt; podle daného formátu, <b>--sort</b> &lt;specifikace&gt; třídí procesy podle dané specifikace, <b>-Z</b> vypíše bezpečnostní kontext SELinuxu</p> <pre>\$ ps -ef   grep sshd</pre> <p>(vyhledá v běžících procesech daný proces)</p> <pre>\$ ps -lt pts/1</pre> <p>(vypíše podrobný seznam běžících procesů na daném terminálu)</p> <pre>\$ ps -p \$\$</pre> <p>(vypíše aktuální proces shellu)</p> <pre>\$ ps -p 1 -o comm=</pre> <p>(vypíše jméno procesu podle jeho PID)</p> <pre>\$ ps -eo pid,ppid,user,%mem,%cpu,comm --sort=-%cpu</pre> <p>(třídí procesy podle vytížení procesoru)</p> <pre>\$ ps -eo pid,ppid,user,%mem,%cpu,comm --sort=-%mem</pre> <p>(třídí procesy podle vytížení paměti)</p>
<p><b>top</b></p>	<p>zobrazí interaktivním a dynamickým způsobem seznam procesů, včetně jejich PID, vlastníka, priority nebo stavu a celkové využití CPU, paměti a swapu, <b>-u</b> &lt;uživatel&gt; procesy daného uživatele, <b>-p</b> &lt;PID&gt; daný proces; interaktivní volby: <b>f</b> zobrazí nabídku položek, které lze přidat jako další sloupec do výstupu příkazu (např. swap, UID či PPID), <b>M</b> seřadí procesy podle využití paměti, <b>P</b> seřadí procesy podle využití procesoru (implicitně), <b>k</b> &lt;PID&gt; ukončí daný proces, <b>q</b> ukončí program</p>

PROCESY	
<b>pidof</b> <jmeno_procesu>	zobrazí PID daného procesu \$ ps -p \$(pidof sshd) (vyhledá v běžících procesech daný proces)
<b>pgrep</b> <vzor>	vypíše PID procesů odpovídajících vzoru, <b>-f</b> odpovídá celému příkazovému řádku (nejen názvu procesu), včetně cesty ke spustitelnému souboru, argumentů a voleb, <b>-l</b> vypíše i jméno procesu, <b>-P</b> <PPID> vypíše procesy nadřazeného procesu, <b>-c</b> vypíše počet odpovídajících procesů, <b>-i</b> ignoruje velikost písmen, <b>-u</b> <uzivatel   UID> procesy daného uživatele \$ pgrep http (vypíše PID httpd procesů)
<b>nice</b> [[ <b>-n</b> ] <priorita>] <prikaz>	spustí proces se změněnou prioritou (hodnotou „nice“), parametr „priorita“ uvádí hodnotu, která se odečte nebo přičte k výchozí hodnotě (implicitně 0), pokud se vynechá, hodnota se automaticky sníží o 10; priorita se udává v rozmezí -20 až 19, čím nižší číslo, tím vyšší priorita, záporné hodnoty smí nastavit pouze root; bez argumentu vypíše aktuální zděděnou prioritu \$ nice -n 15 rm -rf ~/tmp/* (sníží prioritu procesu o 15) # nice -n -15 ls /etc   cpio -ov > /dev/rmt0 (zvýší prioritu procesu o 15)
<b>renice</b> [ <b>-n</b> ] <priorita> <identifikator>	změní prioritu (hodnotu „nice“) běžícího procesu, hodnota priority se zadává v absolutním tvaru v rozmezí -20 až 19; běžný uživatel smí prioritu pouze snižovat, [ <b>-p</b> ] <PID> určí proces, <b>-u</b> <uzivatel> určí procesy daného uživatele, <b>-g</b> <skupina> určí procesy dané skupiny # renice -n 5 987 (nastaví prioritu daného procesu) # renice -n -20 -u root (zvýší prioritu všech procesů uživatele root na maximum) # renice -n 19 -g users (sníží prioritu všech procesů skupiny „users“ na minimum)
<b>Ctrl+c</b>	ukončí běžící proces v popředí
<b>Ctrl+d</b>	ukončí běžící proces, který čte data ze STDIN nebo ukončí současný shell

PROCESY	
<b>kill</b> [<signal>] <PID ...   %cislo_ulohy ...>	<p>(-15) ukončí proces či úlohu odpovídající danému ID standardním způsobem (signál SIGTERM), -9 ukončí proces ihned (signál SIGKILL), -1 ukončí proces, u démonů znovu načte jejich konfigurační soubor (signál SIGHUP), -17 upozorní rodičovský proces na události podřízených procesů (signál SIGCHLD), -19 pozastaví proces (signál SIGSTOP), -18 spustí pozastavený proces (signál SIGCONT), -l vypíše seznam všech signálů, -s &lt;signal&gt; určí signál; signál se zadává číslem nebo jménem (s předponou „SIG“ či bez ní a bez ohledu na velikost písmen)</p> <pre># kill -9 1 # kill -SIGKILL 1 # kill -KILL 1 # kill -sigkill 1 # kill -kill 1</pre> <p>(ukončí ihned daný proces)</p> <pre>\$ kill %3</pre> <p>(ukončí danou úlohu)</p> <pre>\$ kill -s SIGCHLD 1376</pre> <p>(ukončí zombie proces odesláním signálu SIGCHLD rodičovskému procesu)</p>
<b>killall</b> [<signal>] <jmeno_procesu ...>	<p>(-15) ukončí všechny procesy odpovídající danému jménu (vhodné zvláště při více spuštěných procesech jednoho programu) standardním způsobem (signál SIGTERM), -9 ukončí proces ihned (signál SIGKILL), -19 pozastaví proces (signál SIGSTOP), -18 spustí pozastavený proces (signál SIGCONT), -l ignoruje velikost písmen, -l vypíše seznam všech signálů, -s &lt;signal&gt; určí signál, -u &lt;uzivatel&gt; ukončí všechny procesy daného uživatele; signál se zadává číslem nebo jménem (s předponou „SIG“ či bez ní a bez ohledu na velikost písmen)</p> <pre># killall -9 sshd</pre> <p>(ukončí ihned dané procesy)</p>
<b>pkill</b> [<signal>] <vzor>	<p>(-15) ukončí proces odpovídající danému vzoru standardním způsobem (signál SIGTERM), -9 ukončí proces ihned (signál SIGKILL), -19 pozastaví proces (signál SIGSTOP), -18 spustí pozastavený proces (signál SIGCONT), -f odpovídá celému příkazovému řádku (nejen názvu procesu), včetně cesty ke spustitelnému souboru, argumentů a voleb, -i ignoruje velikost písmen, -u &lt;uzivatel   UID&gt; procesy daného uživatele; signál se zadává číslem nebo jménem (s předponou „SIG“ či bez ní a bez ohledu na velikost písmen)</p>
<b>fuser</b> [<soubor ...   zarizeni ...>]	<p>vypíše procesy používající daný soubor či zařízení, -m vypíše procesy používající celý souborový systém, na kterém daný soubor leží, -u vypíše vlastníka procesu, -k ukončí proces, -i žádá o potvrzení, -v podrobný výpis</p> <pre>\$ fuser -kiv /dev/cdrom</pre> <p>(ukončí procesy používající dané zařízení)</p>
<b>strace</b> [<prikaz>]	<p>sleduje systémová volání a signály volané a přijímané procesem, -e &lt;vyraz&gt; sleduje dané systémové volání, -o &lt;soubor&gt; zapíše výstup do souboru, -p &lt;PID&gt; sleduje běžící proces, -c počítá čas, volání a chyby pro každé systémové volání a zobrazí shrnutí při ukončení programu</p>

PROCESY	
<b>ltrace</b> [<prikaz>]	sleduje volání sdílených knihoven a signály volané a přijímané procesem, <b>-e</b> <vyraz> sleduje dané volání knihovny, <b>-o</b> <soubor> zapíše výstup do souboru, <b>-p</b> <PID> sleduje běžící proces, <b>-c</b> počítá čas a volání pro každé volání knihovny a zobrazí shrnutí při ukončení programu, <b>-S</b> navíc zobrazí systémová volání
<b>rpcinfo</b> [<IP_adresa   jmeno_pocitace>]	<b>-p</b> vypíše RPC služby (jejich identifikační číslo, verzi, protokol, port a název) běžící pod správou portmapperu na lokálním či daném počítači; seznam všech RPC programů je uveden v <i>/etc/rpc</i>
<b>clustat</b>	zobrazí stav clusteru
<b>pcs status</b> (platí od RHEL 6)	zobrazí stav clusteru

ÚLOHY	
<b>jobs</b> [%<cislo_ulohy> ...]	vypíše danou úlohu (její číslo, „+“ či „-“ označující poslední nebo předchozí úlohu, status a příkaz spojený s úlohou) v aktuálním shellu, <b>-l</b> včetně PID, <b>-r</b> vypíše pouze běžící úlohy, <b>-s</b> vypíše pouze pozastavené úlohy, <b>-n</b> vypíše pouze změny od předchozího spuštění; bez argumentu vypíše všechny pozastavené úlohy a úlohy běžící na pozadí
<prikaz> &	spustí úlohu na pozadí ve vnořeném shellu
<b>Ctrl+z</b>	pozastaví běžící úlohu v popředí
<b>suspend</b>	pozastaví provádění aktuálního shellu, dokud neobdrží signál SIGCONT, <b>-f</b> pozastaví přihlašovací shell
<b>bg</b> [%<cislo_ulohy> ...]	spustí pozastavenou úlohu na pozadí; bez argumentu spustí poslední pozastavenou úlohu
<b>fg</b> [%<cislo_ulohy>]	spustí pozastavenou úlohu v popředí nebo úlohu z pozadí přesune do popředí; bez argumentu spustí poslední pozastavenou úlohu či přesune poslední spuštěnou úlohu na pozadí do popředí
<b>nohup</b> <prikaz>	spustí daný příkaz tak, že je odolný vůči signálu SIGHUP (běží i po odhlášení uživatele); výstup příkazu se ukládá do souboru „nohup.out“ v pracovním adresáři # nohup find /tmp -name core -print > core.txt & (spustí na pozadí daný příkaz, který běží i po odhlášení uživatele)
<b>watch</b> <prikaz>	spustí příkaz opakovaně v pravidelných intervalech 2 sekund, <b>-n</b> <n> každých <i>n</i> sekund, <b>-d</b> zvýrazní změny na výstupu, <b>Ctrl+c</b> ukončí program \$ watch -n 1 date (zobrazí aktuální čas) \$ watch -d iostat (zobrazí změny na I/O zatížení disků a rychlosti zápisu a čtení dat) \$ watch "ps -eo %cpu,pid,user,args --sort=%cpu   head -11" (zobrazí dynamický seznam 10 procesů, které nejvíce využívají procesor)

ÚLOHY	
<b>wait</b> [<PID   %cislo_ulohy>]	čeká na ukončení daného procesu či úlohy běžící na pozadí aktuálního shellu, pak zanikne; vrátí návratový kód posledního procesu, pokud proces neexistuje, vrátí kód 127; bez argumentu čeká na ukončení všech procesů běžících na pozadí aktuálního shellu a návratový kód je 0 \$ wait \$! && echo "OK" (čeká na ukončení posledního procesu běžícího na pozadí)
<b>sleep</b> <doba>	nastaví časový interval, během něhož nedochází k žádné aktivitě (vhodné zejména v případě, kdy je třeba zajistit určitou časovou prodlevu před spuštěním dalšího příkazu); doba trvání je uvedena číslem a příponou <b>s</b> v sekundách (implicitně), <b>m</b> v minutách, <b>h</b> v hodinách, <b>d</b> ve dnech \$ sleep 10 && echo "Uplynulo 10 sekund." (spustí následující příkaz po uplynutí 10 sekund)
<b>at</b> <cas> [<datum>] [<dalsi_casove_urceni>]	spustí úlohu jednorázově v uvedený čas; příkazy, které se mají provést, se čtou ze STDIN nebo ze souboru, <b>-f</b> <soubor> čte příkazy z daného souboru, <b>-c</b> <cislo_ulohy> zobrazí obsah naplánované úlohy, <b>-d</b> <cislo_ulohy> odstraní naplánovanou úlohu (obdoba příkazu „atrm“), <b>-l</b> vypíše seznam naplánovaných úloh (obdoba příkazu „atq“), <b>-m</b> odešle uživateli email po dokončení úlohy \$ at 20:00 25.06.2007 <- '   \$ at 7am today + 3 weeks <- ' at> mail root < nabidka.txt <- ' at> <b>Ctrl+d</b> (odešle v daný čas uživateli root email s obsahem uvedeného souboru) \$ echo "date > date.txt"   at noon Sunday (spustí v daný čas příkaz, jehož výstup zapíše do uvedeného souboru) \$ at -mf seznam 20:00 25.06.2007 (provede v daný čas seznam příkazů uvedených v souboru a po dokončení úlohy odešle uživateli email)
<b>atq</b>   <b>at -l</b>	vypíše seznam naplánovaných úloh (číslo úlohy, čas spuštění, jméno uživatele, jenž úlohu naplánoval)
<b>atrm</b>   <b>at -d</b> <cislo_ulohy ...>	odstraní naplánovanou úlohu
<b>lpr</b> [<soubor>]	tiskne soubor; není-li soubor uveden, čte ze STDIN, <b>-P</b> <tiskarna> určí tiskárnu, <b>-#</b> <n> určí počet kopií
<b>lpq</b>	vypíše úlohy v tiskové frontě výchozí tiskárny, <b>-P</b> <tiskarna> určí tiskárnu, <b>-a</b> úlohy na všech tiskárnách, <b>-l</b> podrobný výpis
<b>lprm</b> [<cislo_ulohy>   -]	odstraní danou úlohu z tiskové fronty či celou frontu, <b>-P</b> <tiskarna> určí tiskárnu; bez argumentu odstraní současnou úlohu na výchozí tiskárně

ÚLOHY	
<b>crontab</b> [<soubor>]	<p>nastaví existující soubor pro pravidelné spouštění uvedených úloh prostřednictvím démona cron, <b>-e</b> vytvoří či edituje crontabový soubor, <b>-l</b> vypíše obsah crontabového souboru, <b>-r</b> odstraní crontabový soubor, <b>-u</b> &lt;uživatel&gt; nastaví crontabový soubor daného uživatele; v souboru se úlohy uvádí každá na zvláštní řádek, který se u uživatelských souborů skládá z 6 polí oddělených mezerami - 5 pro zadání času (v pořadí minuta, hodina, den v měsíci, měsíc a den v týdnu (ne-so = 0-6), „*“ vyhovuje všem hodnotám, „*/“ &lt;n&gt; každý n-tý časový interval, „-“ odděluje hodnoty téhož pole, „-“ označuje rozsah hodnot) a 1 pro příkaz samotný (pro zadání příkazu se používá absolutní cesta, nejde-li o vestavěný příkaz nebo klíčové slovo shellu, má-li nějaký výstup, odešle se uživateli emailem); prázdný řádek a řádek začínající znakem „#“ se ignoruje; mezi systémové soubory patří /etc/crontab a soubory umístěné v /etc/cron.d, které obsahují 1 pole navíc mezi časovým zadáním a příkazem - uživatele, pod kterým se má daná úloha spustit; soubory uživatelů se nachází ve /var/spool/cron/&lt;uživatel&gt;, a pokud jsou editovány ručně (nikoliv příkazem „crontab -e“), nejsou změny automaticky démonem načteny; oprávnění uživatelé jsou uvedeni v /etc/cron.allow, neoprávnění uživatelé v /etc/cron.deny, existují-li oba soubory, /etc/cron.deny se ignoruje</p> <pre>30 8 * * * /usr/bin/df   /bin/mail admin (spustí úlohu každý den v 8:30) */30 8-16 * * 1-5 /usr/bin/who &gt;&gt; /tmp/users.out (spustí úlohu každých 30 min v pracovní době) 0 22 * * 5 /usr/local/scripts/backup.sh (spustí skript každý pátek ve 22:00) 0 0 1 * * for file in \$(/bin/find /web/logs/*); do &gt; \$file; done (spustí úlohu na začátku každého měsíce)</pre>

From:

<https://prompt.cz/> - **Prompt.cz**

Permanent link:

<https://prompt.cz/procesy-a-ulohy>

Last update: **2025/07/18 12:49**

