

System a terminál

SYSTÉM A TERMINÁL	
uname	[-s] vypíše jméno jádra, -r verzi jádra, -v pořadí a datum kompilace jádra, -p typ procesoru (architekturu), -o jméno operačního systému, -n síťové jméno počítače (nodename), -a všechny informace
lsb_release	vypíše LSB (Linux Standard Base) verzi, -i distribuci Linuxu, -r verzi distribuce, -d distribuci Linuxu a její verzi, -c kódové označení distribuční verze, -s krátký výpis (bez záhlaví), -a všechny informace \$ lsb_release -si (vypíše název distribuce Linuxu)
cat /etc/*release lsb_release -d	vypíše distribuci Linuxu a její verzi
dmidecode	vypíše informace o použitém hardwaru a BIOSu, -s <klicove_slovo> informace o konkrétní položce ze seznamu dostupných klíčových slov, -t <typ> informace o konkrétní položce ze seznamu dostupných typů # dmidecode -s system-product-name (vypíše typ systému (první 4 znaky) a model) # dmidecode -s system-serial-number (vypíše sériové číslo systému) # dmidecode -s system-uuid (vypíše ID systému) # dmidecode -s system-product-name egrep -i 'virtual vm bochs' (vypíše případné informace o virtualizaci systému) # dmidecode -t processor (vypíše informace o procesoru) # dmidecode -t memory (vypíše informace o fyzické paměti) # dmidecode -t baseboard (vypíše informace o základní desce) # dmidecode -t bios (vypíše informace o BIOSu)
lscpu	vypíše informace o procesoru
cat /proc/cpuinfo	vypíše podrobné informace o jednotlivých jádrech procesoru
mpstat [<interval> [<pocet>]]	zobrazí statistiky o využití procesoru, -P {<cislo_CPU[...]> ALL} určí procesor \$ mpstat -P ALL 2 5 (vypíše pětkrát ve dvousekundových intervalech informace o využití procesoru)
sar [<interval> [<pocet>]]	zobrazí statistiky o využití procesoru, -P {<cislo_CPU[...]> ALL} určí procesor, -d statistiky o zatížení disků, -n {<klicove_slovo[...]> ALL} síťové statistiky, -r statistiky o využití paměti, -S statistiky o využití swapu, -f <soubor> čte data ze souboru, -o <soubor> zapíše data do souboru, -h v čitelném formátu \$ sar -P ALL 2 5 (vypíše pětkrát ve dvousekundových intervalech informace o využití procesoru)

SYSTÉM A TERMINÁL	
free	zobrazí celkové množství a využití fyzické paměti a swapu v systému a množství vyrovnávací paměti a mezipaměti používané jádrem v kB (údaje z <code>/proc/meminfo</code>), -m v MB, -h v čitelném formátu, -t souhrn
cat /proc/meminfo	vypíše podrobné informace o využití fyzické paměti a swapu v kB
vmstat [<code><interval></code>] [<code><pocet></code>]	zobrazí statistiky o využití virtuální paměti, swapu, blokových zařízení a procesoru, -d statistiky o zatížení disků, -p <code><diskovy_oddil></code> statistiky o zatížení diskového oddílu, -s výstup ve formátu tabulky, -w výstup v širším formátu \$ <code>vmstat -w 2 5</code> (vypíše pětkrát ve dvousekundových intervalech informace o využití virtuální paměti, swapu, disků a procesoru)
nmon	zobrazí interaktivním a dynamickým způsobem informace o procesoru, paměti, swapu, síti, discích, jádru, souborových systémech, NFS a hlavních procesech
ulimit	[-f] vypíše či nastaví daný limit pro užívání zdrojů systému, -a vypíše všechny nastavené limity, -H nastaví „tvrdý“ limit, -S nastaví „měkký“ limit, -f <code><limit></code> max. velikost vytvářeného souboru v MB, -n <code><limit></code> max. počet otevřených souborů, -t <code><limit></code> max. čas procesoru v sekundách, -u <code><limit></code> max. počet procesů na jednoho uživatele; (trvalé nastavení se provede v <code>/etc/security/limits.conf</code> nebo <code>/etc/security/limits.d/*conf</code>)
setup	spustí nástroj pro konfiguraci systému
getconf [<code><promenna></code>]	zobrazí zadané hodnoty konfigurace systému, -a všechny hodnoty \$ <code>getconf LONG_BIT</code> (zobrazí architekturu systému) \$ <code>getconf _NPROCESSORS_ONLN</code> (zobrazí počet procesorů) \$ <code>getconf -a grep ^NAME_MAX</code> (zobrazí maximální délku názvu souboru v bytech)
getent <code><databaze></code> [<code><klic ...></code>]	zobrazí záznamy z databází (souborů) podporovaných Name Service Switch knihovnamí, nastavených v <code>/etc/nsswitch.conf</code> \$ <code>getent passwd</code> (zobrazí všechny lokální a vzdálené (LDAP či NIS) uživatele v systému) \$ <code>getent services smtp</code> (vypíše název služby a její port a protokol)
locale	zobrazí aktuální nastavení pro každou kategorii jazykové lokalizace, -a vypíše seznam všech dostupných jazykových lokalizací
localectl (platí od RHEL 7)	[status] zobrazí jazykovou lokalizaci systému a rozložení klávesnice (údaje z <code>/etc/locale.conf</code>), set-locale LANG=<lokalizace> nastaví jazykovou lokalizaci systému a rozložení klávesnice # <code>localectl set-locale LANG=cs_CZ.utf8</code> (nastaví českou lokalizaci systému)

SYSTÉM A TERMINÁL	
cal [[[<DD>] <MM>] <RRRR>]	<p>zobrazí kalendář (aktuální měsíc či daný rok nebo daný měsíc v roce), -m týden začíná pondělím, -y celý aktuální rok, -j pořadové dny v roce, -w čísla týdnů, -v vertikální rozložení</p> <pre>\$ cal 12 2005</pre> <p>(zobrazí prosinec 2005)</p> <pre>\$ for year in {2000..2100}; do { cal -mv 04 \$year; echo;} grep -E '^Mo.+[]+3[]+' -C 7; done</pre> <p>(zobrazí ročníky v rozmezí 2000 až 2100, ve kterých připadá 3. dubna na pondělí)</p>
date [<MMDDhhmmRR>]	<p>vypíše či nastaví aktuální datum a čas systému, +<format> vypíše datum a čas v daném formátu (např. %H zobrazí hodinu ve dvojciferném 24hodinovém formátu, %M minutu ve dvojciferném formátu, %S sekundu ve dvojciferném formátu, %d den v měsíci ve dvojciferném formátu, %m měsíc ve dvojciferném formátu, %Y rok ve čtyřciferném formátu, %F ve formátu RRRR-MM-DD, %s v počtu sekund od 1. 1. 1970), -d <specifikace> vypíše datum dle dané specifikace</p> <pre>\$ date +%H:%M-%d.%m-%Y</pre> <p>(vypíše aktuální datum a čas systému ve formátu hh:mm-DD.MM-RRRR)</p> <pre>\$ date -d @123456</pre> <p>(převede čas uvedený v sekundách od 1. 1. 1970 do čitelného formátu)</p> <pre>\$ date -d "+90 days" +%F</pre> <p>(vypíše datum za 90 dní ve formátu RRRR-MM-DD)</p> <pre>\$ TZ='America/New_York' date</pre> <p>(vypíše aktuální datum a čas v daném časovém pásmu)</p>
timedatectl (platí od RHEL 7)	<p>vypíše aktuální datum a čas systému, včetně časového pásma, set-time <RRRR-MM-DD hh:mm:ss> nastaví aktuální datum a čas systému, list-timezones vypíše seznam všech dostupných časových pásem, set-timezone <casove_pasmo> nastaví časové pásmo, set-ntp {true false} povolí či zakáže NTP synchronizaci</p>
tzselect	zvolí časové pásmo v interaktivním režimu
ntpdate <server>	<p>nastaví datum a čas systému podle daného NTP serveru (trvalé nastavení se provede v <i>/etc/ntp.conf</i>)</p> <pre># ntpdate tak.cesnet.cz</pre>
hwclock	<p>-r vypíše hardwarové datum a čas, -s nastaví systémový čas podle hardwarového, -w nastaví hardwarový čas podle systémového, --set --date=<RRMMDD hh:mm:ss> nastaví hardwarový čas na danou hodnotu</p> <pre># hwclock --set --date='780321 22:06:56'</pre>
uptime	<p>vypíše systémový čas, čas od posledního spuštění počítače, počet přihlášených uživatelů a průměrné zatížení systému za posledních 1, 5 a 15 minut, -s čas posledního spuštění systému (obdobu příkazu „who -b“)</p>
who -b uptime -s	vypíše čas posledního spuštění systému

SYSTÉM A TERMINÁL	
dmesg	vypíše zprávy z vyrovnávací paměti jádra, především o hardwarových zařízeních, která jádro detekuje během bootovacího procesu (údaje z <code>/var/log/dmesg</code>), -L barevně zvýrazní důležité zprávy, -T zobrazí datum a čas v čitelném formátu \$ <code>dmesg grep -i usb</code> (vypíše informace o USB zařízeních rozpoznávaných jádrem)
cat /var/log/messages	vypíše všeobecné systémové zprávy, včetně těch, které jsou zaznamenány během spuštění systému
cat /var/log/secure	vypíše zprávy související s bezpečností a ověřováním identity
journalctl (platí od RHEL 7)	vypíše všechny systemd logy uložené journal démonem implicitně v <code>/run/log/journal</code> či <code>/var/log/journal</code> , -b od posledního rebootu (implicitně), -S <code><YYYY-MM-DD [<HH:mm:ss>]></code> od určité doby (může být vyžadováno trvalé ukládání logů), -U <code><YYYY-MM-DD [<HH:mm:ss>]></code> do určité doby (může být vyžadováno trvalé ukládání logů), -n <code><n></code> posledních <code>n</code> záznamů (implicitně 10), -f průběžně zobrazí nově přichozí záznamy, -k pouze zprávy jádra, -u <code><jednotka></code> zprávy pro danou systemd jednotku (službu), --user-unit <code><unit></code> zprávy pro danou jednotku (službu) přihlášeného uživatele, -p <code><priorita></code> zprávy dané nebo vyšší priority, -o <code><vystup></code> určí formát výstupu, -r opačný výstup (nejprve se zobrazí nejnovější záznamy), --disk-usage vypíše celkové využití disku všemi archivovanými a aktivními soubory žurnálu, q ukončí program # <code>journalctl -b -p err</code> (zobrazí všechny systemd logy od posledního rebootu s prioritou „error“ nebo vyšší) # <code>journalctl -u httpd -S yesterday</code> (zobrazí logy týkající se webového serveru od včerejšího dne) # <code>journalctl -ko verbose</code> (zobrazí zprávy jádra v podrobném výstupu) # <code>journalctl --disk-usage</code> (zobrazí celkové využití disku všemi archivovanými a aktivními soubory žurnálu)

SYSTÉM A TERMINÁL	
auditctl	<p>spravuje pravidla audit démona pro zaznamenávání informací o systémových událostech do <code>/var/log/audit/audit.log</code>, -a <seznam,akce> přidá pravidlo systémového volání, -F <specifikace_pole> určí filtr na základě polí záznamů auditu, -C <porovnaní_pole> určí pole k porovnání, -S <systemove_volani> určí systémové volání, -w <soubor adresar> přidá pravidlo pro sledování souboru či adresáře (rekurzivně, v rámci jednoho souborového systému), -p {r w x a} určí typ přístupových práv, při kterém se spustí pravidlo pro sledování („r“ = čtení, „w“ = zápis, „x“ = spuštění, „a“ = změna atributů), -k <klicove_slovo> určí klíčové slovo pro pravidlo auditu, -l vypíše všechna pravidla, -d <seznam,akce> <dalsi_specifikace> odstraní pravidlo systémového volání, -W <soubor adresar> odstraní pravidlo pro sledování souboru či adresáře, -D odstraní všechna pravidla; (trvalé nastavení se provede v <code>/etc/audit/rules.d/audit.rules</code>) <pre># auditctl -a always,exit -S all -F pid=1011</pre> (vytvoří pravidlo pro audit všech systémových volání uskutečněných daným procesem) <pre># auditctl -a always,exit -S openat -F auid=530</pre> (vytvoří pravidlo pro audit souborů otevřených daným uživatelem) <pre># auditctl -w /etc -p wa -k etc_change</pre> (vytvoří pravidlo pro rekurzivní audit změn obsahu a atributů adresáře <code>/etc</code> a přiřadí záznamy ke klíčovému slovu „etc_change“)</p>
aureport	<p>vytvoří souhrnnou zprávu na základě logů audit démona, -a zobrazí zprávy SELinuxu, -au zobrazí pokusy o ověření identity, -l zobrazí pokusy o přihlášení, --failed hlásí pouze neúspěšné události, --success hlásí pouze úspěšné události, -x zobrazí vybrané spustitelné soubory, -ts [<datum>] [<cas>] určí datum/čas zahájení reportu (bez zadání data/času se předpokládá dnes/půlnoc), -te [<datum>] [<cas>] určí datum/čas ukončení reportu (bez zadání data/času se předpokládá dnes/nyní), -i převede číselné hodnoty na jména, --summary zobrazí souhrn jednotlivých elementů, -if <soubor> určí vstupní soubor místo výchozího <code>/var/log/audit/audit.log</code> <pre># aureport -ts today</pre> (vytvoří souhrnnou zprávu za dnešní den) <pre># aureport -i -l --summary</pre> (vytvoří souhrnnou zprávu o všech pokusech o přihlášení) <pre># aureport -x --summary</pre> (vytvoří souhrnnou zprávu o spuštění vybraných spustitelných souborů)</p>

SYSTÉM A TERMINÁL	
ausearch	<p>prohledá logy audit démona na základě zadaných kritérií, -a <ID> určí ID události, -f <soubor adresar> určí jméno souboru či adresáře, -x <spustitelný_soubor> určí jméno spustitelného souboru, -k <klicove_slovo> určí dříve definované klíčové slovo, -m <typ_zpravy> určí typ zprávy, -n <pocitac> určí vzdálený počítač, -p <PID> určí proces, -ts [<datum>] [<cas>] určí datum/čas zahájení reportu (bez zadání data/času se předpokládá dnes/půlnoc), -te [<datum>] [<cas>] určí datum/čas ukončení reportu (bez zadání data/času se předpokládá dnes/nyní), -i převede číselné hodnoty na jména, --format <format> určí formát výstupu, -if <soubor> určí vstupní soubor místo výchozího <code>/var/log/audit/audit.log</code></p> <pre># ausearch -m AVC -ts boot</pre> <p>(vyhledá záznamy SELinuxu od posledního spuštění systému)</p> <pre># ausearch -m AVC -ts recent</pre> <p>(vyhledá záznamy SELinuxu za posledních 10 minut)</p> <pre># ausearch -k etc_change</pre> <p>(vyhledá záznamy na základě dříve definovaného klíčového slova)</p> <pre># ausearch -m USER_LOGIN --format csv > report.csv</pre> <p>(vyhledá záznamy související s přihlašování uživatelů do systému a exportuje výsledky ve formátu CSV)</p>
logger [<zprava>]	<p>vytvoří záznam v systémovém logu, -p <priorita> určí prioritu; bez argumentu zapíše do logu obsah STDIN</p> <pre>\$ logger -p local3.info test</pre> <p>(vytvoří log obsahující řetězec „test“ subsystémem „local3“ s prioritou „info“)</p>
cat /etc/grub.conf cat /etc/lilo.conf	vypíše nastavení zaváděcího (bootovacího) programu pro spuštění OS
cat /etc/default/grub (platí od RHEL 7 - GRUB 2)	vypíše nastavení zaváděcího (bootovacího) programu pro spuštění OS, související skripty se nachází v <code>/etc/grub.d</code>
grub-mkconfig	<p>vytvoří konfigurační soubor pro GRUB na základě dat v <code>/etc/grub.conf</code>, -o <soubor> zapíše vytvořený výstup do souboru namísto STDOUT</p> <pre># grub-mkconfig -o /boot/grub/grub.cfg</pre>
grub2-mkconfig (platí od RHEL 7 - GRUB 2)	<p>vytvoří konfigurační soubor pro GRUB 2 na základě dat v <code>/etc/default/grub</code>, -o <soubor> zapíše vytvořený výstup do souboru namísto STDOUT</p> <pre># grub2-mkconfig -o /boot/grub2/grub.cfg</pre> <p>(vytvoří konfigurační soubor pro GRUB 2 na systému s firmwarem BIOS)</p> <pre># grub2-mkconfig -o /boot/efi/EFI/\${ID}/grub.cfg</pre> <p>(vytvoří konfigurační soubor pro GRUB 2 na systému s firmwarem UEFI)</p>
grub-install <zarizeni>	<p>instaluje GRUB na dané zařízení</p> <pre># grub-install /dev/sda</pre>
grub2-install <zarizeni> (platí od RHEL 7 - GRUB 2)	<p>instaluje GRUB 2 na dané zařízení</p> <pre># grub2-install /dev/sda</pre>

SYSTÉM A TERMINÁL	
efibootmgr	vypíše konfiguraci UEFI Boot Managera, -a aktivuje volbu bootování, -A deaktivuje volbu bootování, -b <cislo_volby> určí volbu bootování, -B odstraní volbu bootování, -o <cislo_volby,cislo_volby...> nastaví pořadí bootování, -v podrobný výpis # efibootmgr (vypíše aktuální volbu bootování, pořadí bootování a všechny dostupné volby bootování) # efibootmgr -b 0006 -B (odstraní danou volbu bootování)
mkinitrd [<obraz_RAM_disku>] <verze_jadra>	vytvoří prvotní obraz RAM disku obsahující základní moduly potřebné k připojení kořenového souborového systému „/“ (teprve potom je možné zavést moduly obsažené v <i>/lib/modules/<verze_jadra></i>), -f přepíše existující soubor „initrd-\$(uname -r).img“, --with=<modul> přidá modul jádra do obrazu RAM disku, -v podrobný výpis # mkinitrd /boot/initrd-\$(uname -r).img \$(uname -r) (obnoví obraz RAM disku) # mkinitrd --with=raid1 /boot/initrd-raid1-\$(uname -r).img \$(uname -r) (nový obraz RAM disku s daným modulem je třeba doplnit do <i>/boot/grub/grub.conf</i> jako nový odstavec – tzn. zkopírovat předchozí část od řádku „title...“ po „initrd...“ a v posledním řádku upravit nový název obrazu; při startu systému je pak možné zvolit, aby naběhl s tímto obrazem)
dracut [<obraz_RAM_souboroveho_systemu>] [<verze_jadra>]] (platí od RHEL 6)	vytvoří prvotní obraz RAM souborového systému obsahující základní moduly potřebné k připojení kořenového souborového systému „/“ (teprve potom je možné zavést moduly obsažené v <i>/lib/modules/<verze_jadra></i>), -f přepíše existující soubor „initramfs-\$(uname -r).img“, -a <modul> přidá dracut modul do obrazu initramfs, -d <modul> přidá modul jádra do obrazu initramfs, --regenerate-all znovu vytvoří všechny obrazy initramfs ve výchozím umístění s verzemi jádra nalezenými v systému, -v podrobný výpis # dracut -f (znovu vytvoří prvotní obraz RAM souborového systému pro aktuální verzi jádra) # dracut -f /boot/initramfs-2.6.32-220.7.1.el6.x86_64.img 2.6.32-220.7.1.el6.x86_64 (znovu vytvoří prvotní obraz RAM souborového systému pro danou verzi jádra)
chroot <korenovy_adresar>	změní kořen souborového systému na nový, daný argumentem, a spustí nový shell (např. při práci v záchranném režimu z externího média) # chroot /mnt/sysimage (připojí se přímo na pevný disk s původní instalací, odkud je možné opravit poškozený systém)
lsmod	vypíše aktuální moduly (ovladače) zavedené v jádru (údaje z <i>/proc/modules</i>)
modinfo <modul ...>	vypíše informace o daném modulu, -d stručný popis, -p vypíše seznam podporovaných parametrů modulu
insmod <modul ...>	přidá modul do jádra, pokud nevyžaduje další závislosti

SYSTÉM A TERMINÁL	
rmmod <modul>	odstraní modul z jádra, pokud nevyžaduje další závislosti, -v podrobný výpis
depmod <modul ...>	vytvoří seznam závislostí daného modulu do <code>/lib/modules/<verze_jadra>/modules.dep</code> , -a všech modulů
modprobe <modul ...>	přidá modul do jádra včetně jeho závislostí, -r odstraní modul, -v podrobný výpis # modprobe usbnet
sysctl [<parametr ...>]	vypíše daný parametr jádra a jeho aktuální hodnotu, -n vypíše pouze hodnotu parametru, -a vypíše všechny parametry jádra a jejich aktuální hodnoty, -w <parametr>=<hodnota> nastaví parametry jádra za běhu (trvalé nastavení se provede v <code>/etc/sysctl.conf</code> nebo <code>/etc/sysctl.d/*</code>), -p <soubor> načte nastavení z daného souboru (implicitně z <code>/etc/sysctl.conf</code>) \$ sysctl vm.swappiness (vypíše hodnotu parametru „vm.swappiness“) # sysctl -w net.ipv4.icmp_echo_ignore_all=1 # echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all (ignoruje požadavek na ping) # sysctl -w net.ipv4.ip_forward=1 # echo 1 > /proc/sys/net/ipv4/ip_forward (povolí IP forwarding) # sysctl -w net.ipv6.conf.all.disable_ipv6=1 # echo 1 > /proc/sys/net/ipv6/conf/all/disable_ipv6 (zakáže IPv6)
systemctl [<prikaz>] (platí od RHEL 7)	vypíše stav všech systemd objektů (jednotek - „units“), show [<jednotka ... uloha ...>] zobrazí vlastnosti dané jednotky, úlohy nebo systemd manažera (není-li zadán žádný argument)
systemctl list-dependencies [<jednotka>] (platí od RHEL 7)	zobrazí ve stromové struktuře seznam všech jednotek („units“), které jsou vyžadovány danou jednotkou; bez argumentu zobrazí potřebné jednotky pro „default.target“ \$ systemctl list-dependencies graphical.target grep "target" (vypíše všechny „target units“, které jsou vyžadovány jednotkou „graphical.target“)
systemd-analyze (platí od RHEL 7)	verify <soubor ...> ověří, že soubor systemd jednotky neobsahuje žádné chyby, security <jednotka ...> analyzuje nastavení zabezpečení jednotky služby systemd, --user spravuje uživatelské systemd soubory (jednotky)
systemctl daemon-reload (platí od RHEL 7)	znovu načte konfiguraci systemd manažera, --user spravuje uživatelské systemd soubory (jednotky)
runlevel	vypíše předchozí a aktuální úroveň běhu systému; pokud úroveň běhu systému nelze určit, vypíše místo toho „N“
who -r	vypíše aktuální úroveň běhu systému a čas jejího spuštění
systemctl list-units -t target (platí od RHEL 7)	vypíše aktuální úroveň běhu systému (aktivní „target units“), --all všechny „target units“

SYSTÉM A TERMINÁL	
init telinit <uroven_behu_systemu>	změní úroveň běhu systému, 0 = vypnutí systému, 1 = jednouživatelský režim bez sítě, 2 = víceuživatelský režim bez sítě, 3 = víceuživatelský režim s připojením k síti, 4 = nevyužito (uživatelsky definovatelné), 5 = úroveň 3 + GUI, 6 = restart systému; konfiguračním souborem procesu init je <code>/etc/inittab</code> , dojde-li k úpravě souboru, jeho opětovné načtení vyvolá příkaz „init q“ # init 5
systemctl isolate <uroven_behu_systemu> (platí od RHEL 7)	změní úroveň běhu systému (aktuální „target unit“), runlevel0.target poweroff.target = vypnutí systému, runlevel1.target rescue.target = jednouživatelský režim bez sítě, runlevel2.target multi-user.target = víceuživatelský režim s připojením k síti, runlevel3.target multi-user.target = víceuživatelský režim s připojením k síti, runlevel4.target multi-user.target = víceuživatelský režim s připojením k síti, runlevel5.target graphical.target = víceuživatelský režim s připojením k síti + GUI, runlevel6.target reboot.target = restart systému # systemctl isolate multi-user.target (změní úroveň běhu systému na víceuživatelský režim s připojením k síti) # systemctl isolate runlevel5.target (změní úroveň běhu systému na víceuživatelský režim s připojením k síti včetně grafického prostředí)
systemctl rescue (platí od RHEL 7)	změní úroveň běhu systému na jednouživatelský režim bez sítě, připojí všechny lokální souborové systémy a spustí důležité systémové služby
systemctl emergency (platí od RHEL 7)	změní úroveň běhu systému na jednouživatelský režim bez sítě, připojí pouze kořenový souborový systém ke čtení a spustí nejdůležitější systémové služby (používá se v případech, kdy nelze spustit rescue režim)
systemctl get-default (platí od RHEL 7)	vypíše výchozí úroveň běhu systému (výchozí „target unit“)
systemctl set-default <uroven_behu_systemu> (platí od RHEL 7)	změní výchozí úroveň běhu systému (výchozí „target unit“) # systemctl set-default graphical.target (změní výchozí úroveň běhu systému na víceuživatelský režim s připojením k síti včetně grafického prostředí)
reboot init 6	ukončí všechny procesy, uloží nezapsaná data na disk, odpojí souborové systémy a restartuje systém
systemctl reboot (platí od RHEL 7)	ukončí všechny procesy, uloží nezapsaná data na disk, odpojí souborové systémy a restartuje systém
halt init 0	ukončí všechny procesy, uloží nezapsaná data na disk, odpojí souborové systémy a zastaví procesor (pohotovostní režim, systém lze bezpečně vypnout manuálně)
systemctl halt (platí od RHEL 7)	ukončí všechny procesy, uloží nezapsaná data na disk, odpojí souborové systémy a zastaví procesor (pohotovostní režim, systém lze bezpečně vypnout manuálně)
poweroff	ukončí všechny procesy, uloží nezapsaná data na disk, odpojí souborové systémy a vypne systém
systemctl poweroff (platí od RHEL 7)	ukončí všechny procesy, uloží nezapsaná data na disk, odpojí souborové systémy a vypne systém

SYSTÉM A TERMINÁL	
shutdown	ukončí všechny procesy, uloží nezapsaná data na disk, odpojí souborové systémy a zastaví procesor, -h <cas> vypne systém, -r <cas> restartuje systém, -c zruší naplánované vypnutí systému, -k <cas> falešné vypnutí systému (pošle pouze varování uživatelům a znemožní nová přihlášení do systému) # shutdown -h now # shutdown -h 22:00 (vypne systém ihned / v daný čas) # shutdown -r +30 "quick reboot" (restartuje systém za 30 min a zobrazí upozornění)
startx	spustí systém X window
xterm	spustí terminál v systému X window, -e <prikaz> určí příkaz, který se má v daném terminálu spustit # xterm -e "tail -f /var/log/secure" & (spustí terminál, ve kterém zobrazí nově příchozí logy)
tty	vypíše jméno terminálu přihlášeného uživatele
stty [<argument>]	vypíše či změní nastavení terminálu přihlášeného uživatele, -a vypíše současné nastavení v čitelném formátu
reset	smaže obsah terminálu a nastaví výchozí hodnoty
clear	smaže obsah terminálu
chvt <n>	změní virtuální terminál v rozsahu 1-12 (obdoba kombinace kláves (Ctrl-)LeftAlt-F<n>)
tmux [<prikaz>]	vytváří a spravuje více terminálových relací v rámci jednoho terminálového okna, Ctrl+b následován " rozdělí aktuální panel vodorovně, Ctrl+b následován % rozdělí aktuální panel svisle, Ctrl+b následován klávesami se šipkami přepíná mezi panely, Ctrl+b následován { přesune aktuální panel doleva, Ctrl+b následován } posune aktuální panel doprava, Ctrl+b následován ! převede aktuální panel na okno, Ctrl+b následován x zavře aktuální panel, Ctrl+b následován c vytvoří nové okno, Ctrl+b následován w zobrazí seznam oken, Ctrl+b následován číslem okna přepíná mezi okny, Ctrl+b následován & zavře aktuální okno, Ctrl+b následován : přepne do příkazového režimu, q ukončí příkazový režim; stavový řádek v dolní části obrazovky zobrazuje informace o aktuální relaci a používá se k zadávání interaktivních příkazů

SYSTÉM A TERMINÁL	
<p>declare typeset [<promenna>[=<hodnota>] ...]</p>	<p>definuje lokální proměnnou a/nebo nastaví její atributy (ve většině případů stačí implicitní deklarace <promenna>=<hodnota>), -p zobrazí atributy a hodnoty každé proměnné, -f zobrazí pouze funkce shellu, -a považuje proměnnou za indexované pole (klíče jsou uspořádaná celá čísla začínající od 0), -A považuje proměnnou za asociativní pole (klíče jsou libovolné řetězce), -i považuje proměnnou za celé číslo (aritmetické vyhodnocení se provede, když je proměnné přiřazena hodnota), -r nastaví proměnnou „pouze ke čtení“ (obdoba příkazu „readonly“), -x exportuje proměnnou do všech podřízených procesů shellu (obdoba příkazu „export“), +<volba> vypne atribut proměnné (kromě „+a“ a „+r“); bez argumentu vypíše všechny proměnné (globální + lokální) a funkce shellu</p> <pre>\$ IndexedArray=(nulla jeden dva) \$ declare -a IndexedArray=(nulla jeden dva) (definuje indexované pole) \$ declare -A AssociativeArray=([system]="linux" [shell]="bash") (definuje asociativní pole)</pre>
<p>readonly declare -r typeset -r [<promenna>[=<hodnota>] ...]</p>	<p>definuje lokální proměnnou „pouze ke čtení“, jejíž hodnoty nelze změnit následným přiřazením a kterou nelze odstranit příkazem „unset“, -f [<funkce>] nastaví funkci „pouze ke čtení“ či vypíše funkce „pouze ke čtení“, -a považuje proměnnou za indexované pole (klíče jsou uspořádaná celá čísla), -A považuje proměnnou za asociativní pole (klíče jsou libovolné řetězce); bez argumentu vypíše všechny proměnné „pouze ke čtení“</p>
<p>export declare -x typeset -x [<promenna>[=<hodnota>] ...]</p>	<p>exportuje lokální proměnnou do všech podřízených procesů shellu (lokální proměnná se trvale nastaví její definicí a exportem v <code>~/.bash_profile</code>, globální proměnná v <code>/etc/profile</code>), -f [<funkce>] exportuje funkci či vypíše exportované funkce, -n <promenna funkce> zruší export proměnné, s volbou „-f“ zruší export funkce; bez argumentu vypíše všechny exportované proměnné</p> <pre>\$ x=y; export x \$ export x=y (exports lokální proměnnou „x“ o hodnotě „y“)</pre>
<p>set</p>	<p>vypíše všechny proměnné (globální + lokální) a funkce shellu, -o vypíše aktuální nastavení voleb shellu, -/+o <volba> změní nastavení voleb shellu („-“ volbu nastaví, „+“ volbu zruší), -x podle místa výskytu ve skriptu spustí skript nebo jeho část v debug režimu, +x ukončí debug režim v místě výskytu ve skriptu</p> <pre>\$ set -o vi (nastaví shellu prostředí editoru „vi“)</pre>
<p>unset <promenna ...></p>	<p>odstraní lokální proměnnou, -f <funkce> odstraní funkci</p> <pre>\$ unset AssociativeArray[system] (odstraní klíč "system" a jeho hodnotu z proměnné asociativního pole)</pre>
<p>printenv [<promenna> ...]</p>	<p>vypíše všechny globální proměnné a jejich hodnoty či hodnoty zadaných proměnných</p>

SYSTÉM A TERMINÁL	
env [<promenna>=<hodnota> <prikaz>]	spustí proces v prostředí dočasně změněném vlastním nastavením proměnných, které se do aktuálního prostředí buď přidávají nebo se z něj odebírají, -u <promenna> odebere proměnnou, -i odebere všechny proměnné (načte se prázdné prostředí); bez argumentu vypíše všechny globální proměnné a jejich hodnoty (obdoba příkazu „printenv“) \$ env LANG=en_EN.UTF-8 date (zobrazí datum v anglickém jazyce)
echo [<text> \$<promenna>]	zobrazí daný text nebo hodnotu proměnné, -e umožní interpretaci escape sekvencí (např. „\n“ = nový řádek, „\t“ = vodorovný tabulátor atd.), -n potlačí nový řádek na konci výstupu; bez argumentu zobrazí prázdný řádek \$ echo "Konec souboru" >> ~/soubor.txt (zapiše daný text na konec souboru) \$ echo \$LANG (zobrazí hodnotu proměnné „LANG“) \$ echo \${AssociativeArray[shell]} (zobrazí hodnotu přiřazenou klíči „shell“ asociativního pole) \$ echo \${AssociativeArray[@]} (zobrazí všechny hodnoty asociativního pole) \$ for val in "\${AssociativeArray[@]}"; do echo \$val; done (zobrazí všechny hodnoty asociativního pole, každou na samostatném řádku) \$ for key in "\${!AssociativeArray[@]}"; do echo \$key \${AssociativeArray[\$key]}; done (zobrazí všechny klíče a jejich hodnoty asociativního pole, každý na samostatném řádku)
printf <format> [<argument>]	zobrazí formátovaný řetězec; specifikace formátu např.: %s řetězec, %d dekadické celé číslo, %% %, \n nový řádek, \t tabulátor \$ printf "Ahoj, jsem %s.\n" \$LOGNAME (vypíše daný text a přihlašovací jméno uživatele) \$ printf "%s\t%s\t%s\n" Nakup Prodej Zisk (vypíše daný text oddělený tabulátory) \$ for i in \$(seq 1 10); do printf "%03d\n" "\$i"; done (vypíše pod sebe čísla 1-10 v trojciferném formátu začínajícím nulou) \$ printf "#%.0s" {1..10} (vypíše daný řetězec 10krát) \$ printf "\U0001f600\n" (zobrazí emoji ikonu)
eval <argument ...>	sloučí dané argumenty a spustí je jako příkaz shellu \$ eval \$(ssh-agent) (spustí příkaz „ssh-agent“ a nastaví proměnné prostředí pro relaci shellu)

SYSTÉM A TERMINÁL	
xargs [<prikaz>]	<p>čte a poté rozdělí větší množství argumentů ze STDIN na menší části tak, aby je bylo možné přiřadit následujícímu příkazu, jenž bude vykonán (příkazům totiž nelze přiřadit neomezený počet argumentů), -0 argumenty jsou ukončeny znakem null místo bílého místa a uvozovky a zpětná lomítka v jejich jménech nemají řídicí význam (tuto volbu však musí podporovat i příkaz poskytující vstup pro „xargs“), -t příkaz se před provedením vypíše na STDERR; bez argumentu spustí příkaz „echo“</p> <pre># find . -name "*core" -print0 xargs -0 -t rm (smaže soubory s názvem „core“) \$ find . -type d xargs chmod 750 (nastaví adresářům daná přístupová práva) \$ find . -atime -1 -type f xargs ls -ltr tail -1 (vypíše naposledy použitý soubor) \$ pgrep ssh xargs (vypíše PID ssh procesů vedle sebe místo pod sebou)</pre>
expr <vyraz>	<p>vyhodnotí matematické a jiné výrazy; jednotlivé argumenty se oddělují mezerami a speciální znaky musí být potlačeny před expanzí shellu</p> <pre>\$ expr 7 + 3 \$ expr "(" 6 - 3 ")" "*" 10 \$ expr length ABCD</pre>
let <argument ...>	<p>vyhodnotí každý argument jako aritmetický výraz</p> <pre>\$ let a=1+2</pre>
bc [<soubor>]	<p>vyhodnotí matematické a jiné výrazy ze zadaného souboru; bez argumentu se spustí v interaktivním režimu</p>
units [<vychozi_jednotka> <cilova_jednotka>]	<p>převede jednotky z jedné stupnice do druhé (a navíc zobrazí převod v opačném směru); bez argumentu se spustí v interaktivním režimu</p> <pre>\$ units 'mile' 'km' (převede míle na kilometry) \$ units '27 inch' 'cm' (převede 27 palců na centimetry)</pre>

SYSTÉM A TERMINÁL	
seq <specifikace>	vypíše posloupnost čísel (po jednom na každý řádek) definovaných horní a případně i dolní mezí nebo inkrementem, -f <format> v daném formátu, -s <retezec> vloží mezi číslice místo nového řádku daný řetězec, -w zarovná řádek dosazením nuly před 1ciferné číslo <pre>\$ seq 2 5</pre> (definovaná dolní i horní mez) <pre>\$ seq 1 .2 3</pre> (definovaná dolní i horní mez včetně inkrementu) <pre>\$ seq -w 15</pre> (číslíce 01-15) <pre>\$ seq -s ':' 10</pre> (číslíce 1-10 oddělené znakem „:“) <pre>\$ seq -f '***%g***' 3</pre> (číslíce 1-3 mezi třemi hvězdičkami) <pre>\$ for i in \$(seq 1 100); do expr \$i "*" \$i; done</pre> (vypíše druhou mocninu čísel 1-100, každou na zvláštní řádek)
shuf [<soubor>]	generuje náhodné permutace (každou na samostatném řádku), -e považuje každý argument za vstupní řádek, -i <m-n> považuje každé číslo daného rozsahu za vstupní řádek, -n <n> určí počet výstupních řádků, -r opakuje výstupní řádky, -o <soubor> určí výstupní soubor; není-li soubor uveden, čte ze STDIN <pre>\$ shuf -e a b c</pre> (generuje náhodné permutace daných znaků) <pre>\$ shuf -n 1 -e Tom Dan Jan</pre> (náhodně vybere jedno ze jmen) <pre>\$ shuf -i 1-9</pre> (generuje náhodná čísla v daném rozsahu) <pre>\$ for n in {01..10}; do echo "\$n:" "\$(shuf -e {01..49} -n 6 tr '\n' ' '); done</pre> (generuje 10krát 6 náhodných čísel v rozsahu 1-49) <pre>\$ shuf -e "\${IndexedArray[@]}"</pre> (generuje náhodné permutace hodnot indexovaného pole)
yes [<retezec ...>]	opakovaně vypisuje „y“ či daný řetězec (každý na samostatném řádku)
timeout <doba> <prikaz>	spustí příkaz s časovým omezením, -s <signal> určí signál, který má být odeslán po uplynutí časového limitu (implicitně „SIGTERM“); doba trvání je uvedena číslem a příponou s v sekundách (implicitně), m v minutách, h v hodinách, d ve dnech <pre>\$ timeout 5 ping prompt.cz</pre> (spustí příkaz „ping“ po dobu 5 sekund)
time <prikaz>	měří dobu provádění daného příkazu <pre>\$ time wget prompt.cz</pre> <pre>\$ time timeout 5s yes</pre>
alias [<vlastni_prikaz[=<prikaz>]]	vytvoří alias pro daný příkaz (trvalé nastavení se provede v <code>~/.bashrc</code>); bez argumentu vypíše všechny stávající aliasy <pre>\$ alias nopaste='curl -F file=@-nopaste.com/a'</pre>
unalias [<vlastni_prikaz ...>]	odstraní daný alias, -a odstraní všechny aliasy

SYSTÉM A TERMINÁL	
script [<soubor>]	zapiše vše, co se od spuštění programu zobrazí na STDOUT do daného souboru, -a přidá výstup na konec existujícího souboru, -c <prikaz> zapiše jen výstup daného příkazu, Ctrl+d ukončí program; bez argumentu zapiše dialog do souboru „typescript“
history [<n>]	vypíše historii posledních <i>n</i> příkazů (implicitně posledních 1000 příkazů), -c smaže historii, -d <n> smaže záznam z historie na dané pozici (řádku), -a přidá novou historii příkazů zadanou od začátku aktuální relace shellu do souboru historie příkazů (<i>.bash_history</i>), -r přidá obsah souboru historie příkazů do aktuálního seznamu historie příkazů
fc [<n> <m n>]	edituje a spustí poslední či vybraný příkaz(y) z historie, -l vypíše očíslovaný seznam posledně spuštěných či vybraných příkazů z historie, -r vypíše seznam příkazů v opačném pořadí, -s provede poslední či vybraný příkaz z historie <pre>\$ fc -l -20 grep cat</pre> (vypíše 20 posledních příkazů obsahujících „cat“) <pre>\$ fc -l 210 250</pre> (vypíše seznam příkazů v daném rozpětí) <pre>\$ fc -s \$!!</pre> (provede předchozí příkaz) <pre>\$ fc -s 560 \$!560</pre> (provede vybraný příkaz)
setenforce <režim>	nastaví režim SELinuxu, je-li povolen (0 = tolerantní (permissive) - bezpečnostní politika není aplikována, pouze se případné varovné zprávy zapisují do <i>/var/log/messages</i> , 1 = vynucující (enforcing) - bezpečnostní politika definující přístup k souborům, portům a procesům je aplikována); trvalé nastavení se provede v <i>/etc/selinux/config</i> ; při změně stavu SELinuxu z „disabled“ na „enforcing“ či „permissive“ nebo obráceně je třeba systém restartovat
getenforce	vypíše aktuální režim SELinuxu („enforcing“, „permissive“, či „disabled“)
sestatus	vypíše aktuální stav SELinuxu, případně jeho režim a použitou politiku, -v podrobný výpis na základě údajů v <i>/etc/sestatus.conf</i>

SYSTÉM A TERMINÁL	
<p>semanage <objekt> <definice></p>	<p>definuje pravidla politiky SELinuxu, objekt fcontext definuje pravidla, která příkaz „restorecon“ používá k nastavení výchozího bezpečnostního kontextu SELinuxu pro soubor či adresář, boolean definuje pravidla pro proměnné, port definuje pravidla pro porty, login definuje pravidla pro uživatele Linuxu, user definuje pravidla pro uživatele SELinuxu, -a přidá (definuje) bezpečnostní kontext, -d smaže bezpečnostní kontext, -m upraví bezpečnostní kontext, -t určí typ kontextu, -l vypíše bezpečnostní kontext, -C s volbou „-l“ vypíše pouze lokální úpravy oproti výchozí politice</p> <pre># semanage fcontext -l (vypíše výchozí bezpečnostní kontext SELinuxu pro soubory a adresáře) # semanage fcontext -at httpd_sys_content_t '/web(/.*)?' (definuje výchozí bezpečnostní kontext SELinuxu pro adresář a jeho obsah) # semanage fcontext -d '/web(/.*)?' (smaže adresáři a jeho obsahu bezpečnostní kontext SELinuxu) # semanage boolean -l grep httpd_enable_homedirs (vypíše, zda je proměnná povolena trvale) # semanage port -l (vypíše typy portů včetně jim přiřazených čísel) # semanage port -at http_port_t -p tcp 81 (povolí webovému serveru poslouchat na TCP portu 81) # semanage login -l (vypíše mapování mezi uživateli Linuxu a uživateli SELinuxu) # semanage user -l (vypíše mapování mezi uživateli SELinuxu a rolemi SELinuxu omezujícími příkazy, které může uživatel provádět) # semanage login -m -s user_u -r s0__default__ (změní výchozí mapování tak, aby mapovalo uživatele Linuxu na uživatele SELinuxu „user_u“) # semanage login -a -s sysadm_u admin (mapuje správce systému na uživatele SELinuxu „sysadm_u“)</pre>
<p>restorecon <soubor ... adresar ...></p>	<p>nastaví (aplikuje) souboru či adresáři výchozí bezpečnostní kontext SELinuxu, -R rekurzivně, -v podrobný výpis</p> <pre># restorecon -Rv /web (rekurzivně nastaví adresáři „/web“ výchozí bezpečnostní kontext SELinuxu)</pre>

SYSTÉM A TERMINÁL	
chcon <kontext> <soubor ... adresar ...>	změní dočasně bezpečnostní kontext SELinuxu souboru či adresáři (změna zůstane zachována do rebootu systému nebo do spuštění příkazu „restorecon“; bezpečnostní kontext se skládá z 5 částí oddělených dvojtečkou - uživatel:role:typ:úroveň:kategorie, přičemž poslední 2 nemusí být v systému definovány; při kopírování se bezpečnostní kontext souboru změní podle nového, nadřazeného adresáře, při přesunu se původní bezpečnostní kontext souboru zachová), -u <uživatel> pro daného uživatele (používá se „root“ pro roota, „user_u“ pro ostatní uživatele a „system_u“ pro procesy), -r <role> pro danou roli (definuje účel daného souboru, procesu nebo uživatele; používá se „object_r“ pro soubory a „system_r“ pro procesy a uživatele), -t <typ> pro daný typ (definuje, které typy procesů mají přístup ke konkrétním typům souborů; „unconfined_t“ znamená, že nespadá pod politiku SELinuxu), --reference <zdroj> nastaví souboru bezpečnostní kontext podle zdrojového souboru, -R rekurzivně, -v podrobný výpis <pre># chcon -Rt httpd_sys_content_t /web</pre> (změní dočasně bezpečnostní kontext adresáře tak, aby jeho obsah byl přístupný pouze pro webový server) <pre># chcon --reference /etc/hosts.allow /etc/hosts.deny</pre> (nastaví druhému souboru bezpečnostní kontext podle prvního souboru)
setsebool <promenna> <stav>	nastaví stav proměnné SELinuxu, -P trvalé nastavení <pre># setsebool -P httpd_enable_homedirs on</pre> (aktivuje danou proměnnou SELinuxu)
getsebool [<promenna>]	vypíše současný stav proměnné SELinuxu, -a vypíše všechny proměnné a jejich stav
seinfo [<vyraz>] [<politika>]	vypíše statistiky politiky SELinuxu, -b seznam proměnných, -r seznam rolí, -t seznam typů (kontextů), -u seznam uživatelů
sealert	zobrazí užitečné informace, které mohou pomoci s řešením problémů se SELinuxem, -a <soubor> vypíše všechny incidenty v souboru, -l <ID> vypíše informace o daném incidentu <pre># sealert -a /var/log/audit/audit.log</pre> (vypíše všechny incidenty v souboru týkající se SELinuxu)

SYSTÉM A TERMINÁL	
virsh <prikaz> [<virtualni_system>] [<argument ...>]	nodeinfo vypíše informace o procesoru a fyzické paměti hostitele (KVM hypervisoru), na kterém běží virtuální systémy (domény), list vypíše aktivní virtuální systémy, domstate <VS> vypíše stav virtuálního systému, start <VS> spustí virtuální systém, shutdown <VS> vypne virtuální systém, reboot <VS> restartuje virtuální systém, dominfo <VS> vypíše základní informace o virtuálním systému, dombklist <VS> vypíše bloková zařízení virtuálního systému, domifaddr <VS> vypíše síťová rozhraní virtuálního systému s jejich IP a MAC adresami, console <VS> otevře konzolu virtuálního systému, net-list vypíše aktivní virtuální sítě, net-start <sit> spustí virtuální síť # virsh list --all (vypíše stav všech virtuálních systémů) # virsh net-start default (spustí „výchozí“ virtuální síť)
exit Ctrl+d	ukončí shell (odhlásí uživatele)
logout	odhlásí uživatele z přihlašovacího shellu

Klávesové zkratky v Bashi	
Doplnění příkazů / názvů souborů:	
Tab	umožňuje rychle doplnit příkazy nebo názvy souborů poté, co byly zadány v dostatečně jedinečném tvaru; pokud zadané znaky nejsou jedinečné, stisknutím klávesy Tab dvakrát zobrazíte všechny dostupné příkazy nebo názvy souborů, které začínají již zadanými znaky
Historie příkazů:	
Ctrl+p Šipka nahoru	vypíše předchozí příkaz v historii
Ctrl+n Šipka dolů	vypíše následující příkaz v historii
Ctrl+r	prohledá historii příkazů podle zadaného vzoru (znovu stiskněte pro další vyhledávání)
Ctrl+o	spustí příkaz nalezený pomocí "Ctrl+r"
Ctrl+g	ukončí vyhledávání v historii příkazů
Řízení procesů:	
Ctrl+c	ukončí běžící proces v popředí
Ctrl+d	ukončí běžící proces, který čte data ze STDIN / ukončí současný shell
Ctrl+z	pozastaví běžící úlohu v popředí
Ctrl+s	pozastaví veškerý výstup na obrazovku (užitečné při spuštění příkazu s dlouhým podrobným výstupem, ale není nutné proces ukončit)
Ctrl+q	obnoví výstup na obrazovku po předchozím zastavení pomocí "Ctrl+s"
Pohyb kurzoru:	
Ctrl+b Šipka vlevo	přesune kurzor o jeden znak doleva
Ctrl+f Šipka vpravo	přesune kurzor o jeden znak doprava
Alt+b	přesune kurzor o jedno slovo doleva
Alt+f	přesune kurzor o jedno slovo doprava

Klávesové zkratky v Bashi	
Ctrl+a	přesune kurzor na začátek příkazového řádku
Ctrl+e	přesune kurzor na konec příkazového řádku
Úprava znaků:	
Ctrl+d Delete	vymaže znak pod kurzorem
Ctrl+h Backspace	vymaže znak před kurzorem
Ctrl+u	vymaže všechny znaky před kurzorem a přidá je do schránky
Ctrl+k	vymaže všechny znaky za kurzorem a přidá je do schránky
Ctrl+l	vymaže celou obrazovku
Ctrl+y	vloží poslední znaky přidané do schránky
Alt+. Esc+.	vloží poslední slovo předchozího příkazu na aktuální pozici kurzoru
Alt+u	převeďte znaky z místa kurzoru po konec aktuálního slova na velká písmena
Alt+l	převeďte znaky z místa kurzoru po konec aktuálního slova na malá písmena
Ctrl+_	vrátí o změnu zpět před posledním stisknutím klávesy

From:

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

Permanent link:

<https://www.prompt.cz/system-a-terminal>

Last update: **2024/09/18 15:32**

