Účty a práva

ÚČTY	
whoami | **echo $USER**	vypíše jméno uživatele odpovídající efektivnímu UID
who am i | **who -m**	vypíše přihlašovací jméno přihlášeného uživatele, jméno terminálu a čas přihlášení
logname	vypíše přihlašovací jméno přihlášeného uživatele
id [<uzivatel ... | UID ...>]	vypíše UID a GID přihlášeného či daného uživatele včetně všech jeho skupin, **-u** efektivní UID, **-g** efektivní GID, **-G** GID všech skupin uživatele, **-n** s volbou „-u“, „-g“ nebo „-G“ vypíše jméno uživatele nebo skupiny místo číselného označení
lid [<uzivatel>]	vypíše skupiny, do nichž je přihlášený či zadaný uživatel přiřazen, **-g** <skupina> vypíše uživatele v dané skupině; příkaz lze použít pouze s právy uživatele root
finger [<uzivatel ...>]	vypíše přihlašovací a skutečné jméno uživatele, jeho domovský adresář, přihlašovací shell, čas posledního přihlášení a informace o doručené poště; bez argumentu zobrazí přihlašovací a skutečné jména právě přihlášených uživatelů, jejich terminál, dobu nečinnosti, čas přihlášení a způsob připojení
lslogins [<uzivatel>]	vypíše informace o všech či zadaných uživatelích v systému – jejich UID, jméno, počet spuštěných procesů, informace o uzamčení účtu, čas posledního přihlášení a komentář (GECOS); u zadaných uživatelů navíc vypíše domovský adresář, přihlašovací shell, primární skupinu, GID, terminál a počítač, **-a** vypíše údaje o poslední změně a vypršení platnosti hesla, **-s** vypíše systémové uživatele, **-u** vypíše běžné uživatele (s UID 1000 a výš)
users	vypíše jména právě přihlášených uživatelů, jejich počet odpovídá počtu současného připojení (údaje z */var/run/utmp*)
who	vypíše jména právě přihlášených uživatelů, jejich terminál a čas přihlášení (údaje z */var/run/utmp*), **-u** včetně PID běžícího procesu, **-q** pouze jména uživatelů a jejich počet, **-m** přihlašovací jméno uživatele, jméno terminálu a čas přihlášení (obdoba příkazu „who am i”)
w [<uzivatel>]	vypíše jména všech právě přihlášených uživatelů či jméno daného přihlášeného uživatele, jejich terminál, způsob připojení, čas přihlášení, dobu nečinnosti, využití procesoru a jméno běžícího procesu; kromě toho zobrazí systémový čas, čas od posledního spuštění počítače, počet přihlášených uživatelů a průměrné zatížení systému za posledních 1, 5 a 15 minut (obdoba příkazu „uptime“)

ÚČTY	
last [<uzivatel ...>]	vypíše časové údaje o přihlášení všech či zadaných uživatelů do systému za poslední období (od vytvoření souboru */var/log/wtmp*) včetně jména terminálu a způsobu připojení, **-f** <soubor> čte údaje z daného souboru, **-n** <n> jen posledních *n* přihlášení, **-s** <cas> od zadaného času, **-t** <cas> do zadaného času, **-d** vypíše jména počítačů při vzdáleném připojení, **-i** vypíše IP adresy při vzdáleném připojení, **-x** vypíše změny úrovní běhu systému `$ last -s -3days` (zobrazí přihlášené uživatele za poslední tři dny) `$ last -s 2021-03-01 -t 2021-03-31` (zobrazí přihlášené uživatele v daném období)
lastb [<uzivatel ...>]	vypíše časové údaje nezdařených pokusů o přihlášení všech či zadaných uživatelů do systému za poslední období (od vytvoření souboru */var/log/btmp*) včetně jména terminálu a způsobu připojení, **-f** <soubor> čte údaje z daného souboru, **-n** <n> jen posledních *n* přihlášení, **-s** <cas> od zadaného času, **-t** <cas> do zadaného času, **-d** vypíše jména počítačů při vzdáleném připojení, **-i** vypíše IP adresy při vzdáleném připojení
lastlog	vypíše seznam všech uživatelů v systému a čas jejich posledního přihlášení včetně jména terminálu (údaje z */var/log/lastlog*), **-u** <uzivatel> záznamy daného uživatele, **-t** <n> údaje uživatelů přihlášených během posledních *n* dní
faillog	vypíše záznamy o neúspěšném přihlášení všech uživatelů (údaje z */var/log/faillog*), **-a** všechny záznamy, **-u** <uzivatel> záznamy daného uživatele, **-m** <n> nastaví max. počet chybných přihlášení, **-r** vynuluje počítadlo chybných přihlášení; používá modul „pam_tally“ `# faillog -u kuba -r` (vynuluje počítadlo chybných přihlášení daného uživatele)
pam_tally2 (platí od RHEL 6)	vypíše záznamy o neúspěšném přihlášení všech uživatelů (údaje z */var/log/tallylog*), **-u** <uzivatel> záznamy daného uživatele, **--reset** vynuluje počítadlo chybných přihlášení; používá modul „pam_tally2“ `# pam_tally2 --reset -u jan` (vynuluje počítadlo chybných přihlášení daného uživatele)
faillock (platí od RHEL 6)	vypíše záznamy o neúspěšném přihlášení všech uživatelů (údaje z */var/run/faillock/**), **--user** <uzivatel> záznamy daného uživatele, **--reset** vynuluje počítadlo chybných přihlášení; používá modul „pam_faillock“ `# faillock --user tom --reset` (vynuluje počítadlo chybných přihlášení daného uživatele)

ÚČTY	
authconfig <volby>	**--update** aktualizuje PAM konfiguraci podle daných voleb (změní */etc/pam.d/*-ac* konfigurační soubory), **--updateall** obnoví PAM konfiguraci podle nastavení v */etc/sysconfig/authconfig*, **--test** vypíše současnou PAM konfiguraci, **--savebackup=**<adresar> zálohuje PAM konfigurační soubory, **--restorebackup=**<adresar> obnoví PAM konfigurační soubory `# authconfig --enablefaillock --faillockargs="deny=5 fail_interval=900 unlock_time=3600" --update` (povolí a nastaví modul „pam_faillock“ s limitem 5 neúspěšných pokusů o přihlášení v 15minutovém intervalu a automaticky odemkne uzamčené účty po 60 minutách)
useradd <uzivatel>	vytvoří uživatele včetně jeho domovského adresáře */home/<uzivatel>* (zkopíruje sem obsah adresáře */etc/skel*), mailové schránky */var/spool/mail/<uzivatel>* a přiřadí mu primární skupinu téhož názvu; při vytváření nového účtu se vychází z nastavení uvedených v */etc/default/useradd* a */etc/login.defs*, **-D** vypíše výchozí hodnoty, **-d** <adresar> určí domovský adresář namísto výchozího, **-g** <skupina | GID> přiřadí existující skupinu jako primární, **-G** <skupina> přiřadí uživatele do dalších doplňkových skupin oddělených od sebe čárkou, **-u** <UID> nastaví UID (jinak systém přidělí nejbližší volné), **-o** nastaví duplicitní UID (platí pouze s volbou „-u“), **-r** vytvoří systémový účet (s UID v rozsahu 201–999, s neomezenou platností hesla a bez domovského adresáře), **-s** <shell> nastaví přihlašovací shell, **-e** <RRRR-MM-DD> nastaví konec platnosti účtu, **-f** <DD> nastaví životnost účtu v řádu dní po skončení platnosti hesla, **-c** <komentar> poskytne další informace o uživateli (tzv. pole GECOS v */etc/passwd*), **-Z** <SELinux_uzivatel> mapuje uživatele na daného uživatele SELinuxu `# useradd -c "Jan Novak" -G admins jan` (vytvoří uživatele „jan“ s komentářem „Jan Novák“ a doplňkovou skupinou „admins“)
userdel <uzivatel>	odstraní uživatele, **-r** včetně domovského adresáře a mailové schránky, **-f** i když je právě přihlášen, včetně domovského adresáře a mailové schránky, **-Z** včetně mapování na uživatele SELinuxu
usermod <uzivatel>	změní atributy uživatele; používají se stejné volby jako u příkazu „useradd“, navíc **-a** s volbou „-G“ přiřadí uživatele do dalších doplňkových skupin oddělených od sebe čárkou, aniž by bylo nutné uvádět i všechny dříve definované skupiny (samotná volba „-G“ totiž vždy definuje všechny platné doplňkové skupiny od začátku, čímž přepíše předchozí nastavení), **-l** <novy_uzivatel> přejmenuje uživatele, **-L** uzamkne účet (vloží znak „!“ před šifrované heslo), **-U** odemkne účet (odstraní znak „!“ před šifrovaným heslem) `# usermod -c "" jan` (smaže komentář u daného uživatele) `# usermod -l honza -d /home/honza jan` (přejmenuje uživatele „jan“ na „honza“ a změní jeho domovský adresář na „/home/honza“)

ÚČTY	
chfn [<uzivatel>]	změní komentář (GECOS) v */etc/passwd* přihlášeného či zadaného uživatele, **-f** <jmeno> skutečné jméno, **-p** <cislo> služební telefonní číslo, **-h** <cislo> soukromé telefonní číslo; bez argumentu se spustí v interaktivním režimu (none = prázdné pole)
chsh [<uzivatel>]	[**-s** <shell>] změní přihlašovací shell přihlášeného či zadaného uživatele, **-l** vypíše seznam dostupných shellů z */etc/shells*; bez argumentu se spustí v interaktivním režimu
chage <uzivatel>	změní nastavení platnosti účtu a hesla uživatele, **-d** <DD> určí počet dní od 1. 1. 1970, kdy bylo heslo naposledy změněno, **-E** <RRRR-MM-DD> nastaví konec platnosti účtu („-1" = platnost účtu není omezena), **-I** <DD> nastaví počet dní nečinnosti po vypršení platnosti hesla před uzamčením účtu, **-l** vypíše údaje o platnosti účtu a hesla, **-m** <DD> nastaví min. platnost hesla v řádu dní („0" = uživatel smí heslo změnit kdykoliv), **-M** <DD> nastaví max. platnost hesla v řádu dní („-1" = platnost hesla není omezena), **-W** <DD> nastaví počet dní, během nichž je uživatel varován před koncem platnosti hesla; nezadají-li se v příkazu žádné volby, spustí se v interaktivním režimu; výchozí nastavení platnosti hesel uživatelů jsou uvedeny v */etc/login.defs* `# chage -d 0 jakub` (změní konec platnosti hesla daného uživatele a vyzve ho k jeho změně při prvním přihlášení)
passwd [<uzivatel>]	nastaví nebo změní heslo přihlášeného či daného uživatele, **--stdin** čte heslo ze STDIN (roury), **-d** nastaví účet bez hesla, **-n** <DD> určí min. platnost hesla v řádu dní, **-x** <DD> určí max. platnost hesla v řádu dní, **-w** <DD> určí počet dní k varování uživatele před koncem platnosti hesla, **-l** uzamkne účet (vloží znaky „!!" před šifrované heslo), **-u** odemkne účet, **-S** <uzivatel> vypíše informace o nastavení hesla uživatele (stav hesla: „PS" = heslo nastaveno, „NP" = žádné heslo, „LK" = heslo uzamčeno, datum poslední změny hesla, min. a max. platnost hesla v řádu dní, varovací období před vypršením hesla a doba mezi koncem platnosti hesla a uzamčením účtu v řádu dní); výchozí nastavení platnosti hesel uživatelů jsou uvedeny v */etc/login.defs* `# for user in $(awk -F : '{print $1}' /etc/passwd); do passwd -S $user | grep LK; done` (vypíše uživatele se zamčenými účty)
mkpasswd	vytvoří náhodné heslo, **-l** <n> určí délku hesla (implicitně 9 znaků), **-C** <n> určí min. počet velkých písmen (implicitně 2), **-c** <n> určí min. počet malých písmen (implicitně 2), **-d** <n> určí min. počet číslic (implicitně 2), **-s** <n> určí min. počet speciálních znaků (implicitně 1)

ÚČTY	
chpasswd <uzivatel>**:**<heslo>	změní heslo daného uživatele a zašifruje ho algoritmem uvedeným v */etc/login.defs*, **-c** {NONE|DES|MD5|SHA256|SHA512} určí odlišný šifrovací algoritmus, **-e** značí, že nově zadané heslo je v šifrované podobě (jinak se uvádí v čistém textu) `# for user in $(awk -F ":" '{if (length($2) > 2 && $2 !~ /^(!!)?(\$[1256]\$)/) print $1":"$2 }' /etc/shadow); do echo "$user" | chpasswd -c SHA512; done` (zašifruje nezašifrovaná hesla všech uživatelů)
cat /etc/passwd	vypíše existující lokální uživatele, jejich šifrované heslo (vložením znaku „*" dojde k uzamčení účtu) či znak „x" (heslo je v */etc/shadow*), UID, primární GID, komentář (GECOS), domovský adresář a přihlašovací shell `$ awk -F ":" '{if ($7 ~ /.+(sh\|bash\|ksh\|zsh)$/ && ($3 >= 500)) print $1}' /etc/passwd` (vypíše běžné uživatele v systému – s UID 500 a výš) `$ awk -F ":" '{if ($4 == 3000) print $1}' /etc/passwd` (vypíše všechny uživatele, jejichž primární skupina má GID 3000)
cat /etc/shadow	vypíše existující lokální uživatele, jejich šifrované heslo (pokud je pole prázdné, účet je bez hesla; vložením znaku „*", „!" nebo „!!" před heslo dojde k uzamčení účtu; příkaz „useradd" implicitně vytváří zamčené účty – tj. místo hesla obsahují pouze „!!"), čas poslední změny hesla v řádu dní (počítáno od 1. 1. 1970), min. platnost hesla v řádu dní („0" = uživatel smí heslo změnit kdykoliv), max. platnost hesla v řádu dní („-1" = platnost hesla není omezena), varovací období před vypršením hesla v řádu dní, počet dní od vypršení hesla do zablokování účtu a čas zablokování účtu v řádu dní (počítáno od 1. 1. 1970)
groupadd <skupina>	vytvoří novou skupinu, **-g** <GID> nastaví GID (jinak systém přidělí nejbližší volné), **-o** nastaví duplicitní GID (platí pouze s volbou „-g"), **-r** vytvoří systémovou skupinu (s GID v rozsahu 201–999) `# groupadd -g 350 -r nexus` (vytvoří systémovou skupinu „nexus" s GID 350)
groupdel <skupina>	smaže danou skupinu (nelze smazat primární skupinu existujícího uživatele, ten musí být odstraněn první)
groupmod <skupina>	změní atributy skupiny, stejné přepínače jako u příkazu „groupadd", navíc existuje **-n** <nova_skupina> přejmenuje skupinu
groups | **id -nG** [<uzivatel>]	vypíše skupiny, do nichž je přihlášený či daný uživatel přiřazen
newgrp <skupina>	přihlásí uživatele do jedné ze skupin povolených v */etc/group*; bez argumentu přidělí GID jeho primární skupiny (využití zejména při vytváření nových souborů)

ÚČTY	
cat /etc/group	vypíše existující lokální skupiny, jejich šifrované heslo (vložením znaku „*" dojde k uzamčení účtu) či znak „x" (heslo je v */etc/gshadow*), GID a explicitní uživatele oddělené od sebe čárkou `$ awk -F ":" '/^admin/{print $3}' /etc/group` (vypíše GID skupiny „admin") `$ awk -F ":" '{if ($3 == 3000) print $4}' /etc/group` (vypíše všechny explicitní uživatele skupiny, jejíž GID je 3000)
cat /etc/gshadow	vypíše existující lokální skupiny, jejich šifrované heslo (vložením znaku „*" dojde k uzamčení účtu) či znak „!" (účet bez hesla), administrátory a neimplicitní uživatele oddělené od sebe čárkou
vipw	edituje */etc/passwd*
vigr	edituje */etc/group*
pwconv	vytvoří soubor */etc/shadow* na základě údajů z */etc/passwd* a */etc/login.defs*, čímž zajistí bezpečné uložení hesel uživatelských účtů
pwunconv	smaže soubor */etc/shadow* (opak příkazu „pwconv")
grpconv	vytvoří soubor */etc/gshadow* na základě údajů z */etc/group* a */etc/login.defs*, čímž zajistí bezpečné uložení hesel skupinových účtů
grpunconv	smaže soubor /etc/gshadow (opak příkazu „grpconv")
pwck	porovná správnost obsahu souborů */etc/passwd* a */etc/shadow*, případné nesrovnalosti je uživatel vyzván opravit, **-r** pouze vypíše chyby, **-s** třídí zápis podle UID
grpck	porovná správnost obsahu souborů */etc/group* a */etc/gshadow*, případné nesrovnalosti je uživatel vyzván opravit, **-r** pouze vypíše chyby, **-s** třídí zápis podle GID

PRÁVA	
umask [<prava>]	vypíše či nastaví implicitní práva pro nově vytvořené soubory a adresáře v pořadí vlastník - skupina - ostatní v číselném vyjádření (osmičkové soustavě), avšak čísla uvádí práva, jež budou odebrána od stanovené systémové hodnoty 666 pro soubory a 777 pro adresáře, **-S** symbolické vyjádření; (trvalé nastavení se provede v *~/.bashrc* a *~/.bash_profile*, výchozí globální hodnota je 002 pro běžné uživatele a 022 pro uživatele root v */etc/profile* a */etc/bashrc*) `$ umask 0027` | `$ umask 027` | `$ umask 27` (vlastník má všechna práva, skupina práva pro čtení a vstup do adresáře a ostatní nemají práva žádná)

PRÁVA	
chmod <prava> <soubor ... | adresar ...>	nastaví uživatelům přístupová práva k souboru či adresáři 1) v symbolickém vyjádření: – nejdříve se definují uživatelé (**u** = vlastník, **g** = skupina, **o** = ostatní, **a** = všichni), následuje operátor (**+** pro přidání, **-** pro odebrání a **=** pro nastavení práv) a specifikace práv (**r** = čtení souboru / výpis obsahu adresáře (pouze názvy jmen souborů či adresářů), **w** = zápis do souboru / zápis do adresáře (vytváření, mazání a přejmenování jakýchkoliv souborů či adresářů), **x** = spuštění souboru / vstup do adresáře a zpřístupnění jeho obsahu pro čtení a zápis, **X** = vstup do adresáře a zpřístupnění jeho obsahu pro čtení a zápis, **s** = SUID či SGID bit, **S** = „s“ a chybějící „x“, **t** = sticky bit, **T** = „t“ a chybějící „x“) `$ chmod +x script.sh` (nastaví práva ke spuštění souboru pro všechny uživatele) `$ chmod o= .` (odebere ostatním všechna práva k pracovnímu adresáři) `$ chmod ug=rw,o-w text.txt` (nastaví práva ke čtení a zápisu pro vlastníka a skupinu a odebere práva zápisu pro ostatní) 2) v číselném vyjádření (osmičkové soustavě): – v pořadí (speciální atribut) - vlastník - skupina - ostatní (**4** = právo čtení souboru / výpis obsahu adresáře (pouze názvy jmen souborů či adresářů), **2** = zápis do souboru / zápis do adresáře (vytváření, mazání a přejmenování jakýchkoliv souborů či adresářů), **1** = spuštění souboru / vstup do adresáře a zpřístupnění jeho obsahu pro čtení a zápis), hodnoty se sčítají `# chmod 640 /etc/crontab` (nastaví práva ke čtení a zápisu pro vlastníka a ke čtení pro skupinu) u obou variant lze využít volby **-R** pro rekurzivní nastavení a **-c** pro výpis souborů, jejichž práva se mění; adresář musí mít vždy přiděleno právo vstupu `# chmod -R go-w /var/www/html` (nastaví rekurzivně práva pro všechny soubory a adresáře v dané cestě) speciální atributy se týkají zejména spustitelných souborů (programy a skripty) či adresářů a mají tyto hodnoty: **4** = SUID bit (spuštěný proces běží s právy vlastníka souboru, nikoliv uživatele, který ho spustil), **2** = SGID bit (proces běží s právy skupiny vlastníků souboru; je-li SGID bit nastaven u adresáře, zajistí, že nově vytvořený obsah bude vlastnit stejná skupina vlastníků, která vlastní daný adresář), **1** = sticky bit (používá se u adresářů, jejichž obsah může mazat či přejmenovat jen daný vlastník souboru nebo adresáře, nikoliv každý, kdo má práva vstupu a zápisu v daném adresáři) `# chmod 4755 /usr/bin/passwd` (nastaví SUID bit pro daný soubor) `# chmod 2770 /web` (nastaví SGID bit pro daný adresář) `# chmod +t /usr/local/tmp` (nastaví sticky bit pro daný adresář)

PRÁVA	
setfacl <volba> [[<uzivatel>]**:**[<prava>]] <soubor ... | adresar ...>	**-m** nastaví přístupová práva ACL k souboru či adresáři podle daných voleb (**u:**[<uzivatel | UID>] pro daného uživatele, není-li uveden, nastavení platí pro vlastníka souboru či adresáře, **g:**[<skupina | GID>] pro danou skupinu, není-li uvedena, nastavení platí pro skupinu vlastníků souboru či adresáře, **o** pro ostatní uživatele, **d:** zajistí dědění ACL práv z adresáře na jeho nově vytvořený obsah, **m:** nastaví masku – určí maximální možná oprávnění pro všechny jmenované uživatele a skupiny), **-x** smaže přístupová práva ACL k souboru či adresáři podle daných voleb (**u:**<uzivatel | UID> pro daného uživatele, **g:**<skupina | GID> pro danou skupinu), **-b** zruší všechna přístupová práva ACL k souboru či adresáři, **-R** rekurzivně, **--set-file** <soubor | adresar> nastaví přístupová práva ACL podle daného souboru či adresáře `$ setfacl -m d:u::rwx,g::rx,o:000 ./projekty` (nastaví přístupová práva ACL k danému adresáři i nově vytvořenému obsahu) `$ setfacl -m u:kuba:rw ./projekty/kuba.txt` (nastaví přístupová práva ACL k danému souboru pro daného uživatele) `$ setfacl -x u:kuba ./projekty/kuba.txt` (smaže přístupová práva ACL k danému souboru pro daného uživatele) `$ setfacl -bR ./projekty` (zruší rekurzivně všechna přístupová práva ACL v dané cestě) `# setfacl -m u::rwx,g::rx,o::rx /bin/chmod` (nastaví přístupová práva ACL k danému souboru pro vlastníka, skupinu a ostatní) `$ getfacl soubor1 | setfacl --set-file - soubor2` (nastaví souboru „soubor2“ stejná ACL práva jako má „soubor1“)
getfacl <soubor ... | adresar ...>	vypíše přístupová práva ACL k souboru či adresáři i pro jednotlivé uživatele a skupiny (jsou-li nastavena), **-n** zobrazí UID a GID místo názvu účtu, **-R** rekurzivně, **-s** vynechá soubory s běžným nastavením práv
chattr <operator><atribut> <soubor ... | adresar ...>	nastaví atributy daného souboru či adresáře v souborovém systému ext2, ext3 či ext4; operátor **+** přidá, **-** odebere a **=** nastaví atribut; atribut **a** zakáže smazání a úpravu souboru (dokonce i uživateli root), povolí jen přidání nových dat na jeho konec, **d** zakáže zálohu souboru programem „dump“, **i** zakáže smazání a jakoukoliv úpravu souboru (dokonce i uživateli root); **-R** rekurzivně `# chattr +i /etc/inittab` (přidá atribut danému souboru)
lsattr [<soubor ... | adresar ...>]	vypíše atributy obsahu pracovního adresáře či zadaného souboru nebo obsahu zadaného adresáře v souborovém systému ext2, ext3 či ext4, **-a** zobrazí i skryté soubory, **-d** samotný adresář bez obsahu, **-R** rekurzivně

PRÁVA	
chown [<vlastnik>][:[<skupina>]] <soubor ... | adresar ...>	změní vlastníka a/nebo skupinu vlastníků souboru či adresáře, **-R** rekurzivně, **-c** vypíše soubory, jichž se změna týká; následuje-li za označením uživatele (jeho jménem či UID) tečka nebo dvojtečka a označení skupiny (její jméno či GID), změní se zároveň i skupina vlastníků souboru; chybí-li označení skupiny (*chown user: /tmp /var/tmp*), nastavi se primární skupina uživatele; zadá-li se pouze tečka či dvojtečka a označení skupiny (*chown :group /tmp /var/tmp*), změní se jen skupina vlastníků souboru (obdoba příkazu „chgrp")
chgrp <skupina> <soubor ... | adresar ...>	změní skupinu vlastníků souboru či adresáře; označením skupiny se rozumí její jméno či GID, **-R** rekurzivně, **-c** vypíše soubory, jichž se změna týká
su [<uzivatel>]	přihlásí se na účet roota (správce systému) či daného uživatele (změní efektivní UID a GID), **-** | **-l** včetně nastavení uživatelského prostředí (inicializuje proměnné „HOME", „SHELL", „USER", „LOGNAME" a „PATH"), **-c** <prikaz> pod jiným uživatelem provede pouze daný příkaz

PRÁVA	
sudo [<prikaz>]	povolí oprávněnému uživateli spustit příkaz s právy roota či jiného uživatele za použití vlastního hesla; tento uživatel musí být uveden v */etc/sudoers(.d/*)* v pořadí <uzivatel> <pocitac> = [([<efektivni_uzivatel>][:<efektivni_skupina>])] [<tag>:] <prikaz> (v absolutním tvaru); na začátku souboru lze velkým písmem definovat aliasy zastupující oprávněné uživatele, efektivní uživatele, počítače a příkazy, přičemž výraz „ALL“ zastupuje jakoukoliv hodnotu v uvedených položkách: *kuba ALL = (root) /bin/mount -t iso9660 /dev/cdrom /mnt/cdrom, NOPASSWD: /bin/umount /mnt/cdrom* (kuba smí jako root připojit a bez hesla odpojit CD-ROM mechaniku) *miro localhost = /bin/su [!-]*, !/bin/su *root** (miro se smí na lokálním počítači přihlásit pod jakýmkoliv uživatelem kromě root, bez nastavení jeho prostředí) *%admin ALL = SERVICES, PROCESSES, STORAGE* (členové skupiny „admin“ smí na jakémkoliv počítači spustit příkazy zastoupené danými aliasy) *%osadmin ALL=(ALL) ALL* (členové skupiny „osadmin“ smí na jakémkoliv počítači spustit jakýkoliv příkaz) **-b** spustí daný příkaz na pozadí, **-l** vypíše, zda a případně v jakém rozsahu je přihlášený uživatel oprávněn použít „sudo“, **-i** přihlásí se na účet roota, **-u** <uzivatel> spustí příkaz jako jiný uživatel než root, **-g** <skupina> spustí příkaz s oprávněními dané primární skupiny; soubor */etc/sudoers* edituje pouze root příkazem „visudo“; použití příkazu „sudo“ se zapisuje do */var/log/secure* `$ sudo vi /etc/fstab` (edituje soubor s právy roota) `$ sudo bash -c "cd /home; du -s * | sort -rn > usage"` (spustí příkazy ve vnořeném shellu s právy roota, čímž zajistí fungování příkazu „cd“ a přesměrování do souboru) `$ sudo su - root -c /bin/bash` (spustí shell s právy roota)
sudoedit <soubor>	povolí oprávněnému uživateli editovat soubor s právy roota či jiného uživatele za použití vlastního hesla, **-u** <uzivatel> edituje soubor jako jiný uživatel než root, **-g** <skupina> edituje soubor s oprávněními dané primární skupiny; pokud soubor neexistuje, vytvoří ho
visudo	edituje */etc/sudoers*, **-c** kontroluje správnost souboru, **-f** <soubor> určí alternativní soubor místo */etc/sudoers*