Zařízení a souborový systém

ZAŘÍZENÍ A DISKOVÉ ODDÍLY	
cat /proc/devices	vypíše seznam znakových a blokových zařízení seřazených podle jejich hlavního (major) čísla

lshw	vypíše podrobné informace o nalezeném hardwaru a jeho konfiguraci, **-C** <trida> zobrazí pouze danou třídu hardwaru, **-short** zobrazí hardwarové cesty, zařízení, třídy a popis hardwaru
	`# lshw -C system`
	(zobrazí informace o systému)
	`# lshw -C bus`
	(zobrazí informace o základní desce)
	`# lshw -C processor`
	(zobrazí informace o procesoru)
	`# lshw -C disk`
	(zobrazí informace o discích)
	`# lshw -C memory`
	(zobrazí informace o fyzické paměti)
	`# lshw -C network`
	(zobrazí informace o síťových rozhraních)
	`# lshw -C display`
	(zobrazí informace o grafické kartě)
	`# lshw -C multimedia`
	(zobrazí informace o zvukové kartě)

lspci	vypíše informace o zařízeních připojených na PCI sběrnici, **-k** vypíše ovladače a moduly jádra používané pro konkrétní zařízení, **-v** podrobný výpis
	`$ lspci | grep -i ethernet`
	(zobrazí síťová rozhraní)
	`$ lspci | grep -i fibre`
	(zobrazí fibre channel adaptéry)
	`# lspci -v`
	(zobrazí podrobné informace o všech zařízeních připojených na PCI sběrnici)

lsusb	vypíše informace o zařízeních připojených na USB sběrnici, **-v** podrobný výpis
lsscsi	vypíše informace o zařízeních připojených na SCSI sběrnici, **-s** velikost disků, **-u** LUN (identifikátor logické jednotky), **-w** WWN (identifikátor zařízení – obdoba MAC adresy), **-v** podrobný výpis `$ lsscsi | grep tape` (vypíše seznam všech SCSI páskových zařízení)
rescan-scsi-bus.sh	skenuje všechny SCSI sběrnice, **-a** zkontroluje i nově přidaná zařízení

ZAŘÍZENÍ A DISKOVÉ ODDÍLY	

lsblk [<zarizeni ... | diskovy_oddil ... | LV ...>]	vypíše informace o všech či zadaných blokových zařízeních ve stromové struktuře (název zařízení, hlavní (major) a vedlejší (minor) číslo zařízení, zda je zařízení odnímatelné, velikost zařízení, zda je zařízení pouze ke čtení, typ zařízení a přípojný bod), **-a** včetně prázdných zařízení, **-f** typ souborového systému, název svazku a UUID, **-m** vlastníka, skupinu a práva, **-p** absolutní cestu k zařízení, **-S** pouze SCSI zařízení, **-o** <seznam> určí informace (sloupce), které se mají zobrazit, **-l** výstup ve formě seznamu
	`$ lsblk -o NAME,SIZE`
	(vypíše název a velikost všech blokových zařízení)
	`$ lsblk | grep mpath | sort -u`
	(vypíše seřazený seznam multipath zařízení)

blkid [<zarizeni ... | diskovy_oddil ... | LV ...>]	vypíše atributy všech či zadaných blokových zařízeních (název zařízení, UUID, název svazku, velikost bloků a typ souborového systému), **-L** <LABEL> vypíše diskový oddíl či logický svazek, který používá dané označení, **-U** <UUID> vypíše diskový oddíl či logický svazek, který používá dané UUID
multipath [<zarizeni>]	detekuje více cest ke vzdáleným diskům, **-ll** zobrazí současnou multipath topologii, **-r** znovu načte všechna mapování vzdálených disků, **-t** vypíše výchozí hodnoty nastavení, **-f** smaže mapování daného nepoužívaného vzdáleného disku, **-F** smaže mapování všech nepoužívaných vzdálených disků
mpathconf [<prikaz ...>]	zobrazí device mapper multipath konfiguraci, **--enable** povolí multipathing, **--disable** zakáže multipathing
hdparm <zarizeni ... | diskovy_oddil ...>	vypíše či nastaví parametry daného SATA/IDE zařízení, **-I** vypíše podrobné informace o disku `# hdparm -r0 /dev/sdb*` (odstraní ochranu proti zápisu z USB zařízení)
blockdev <prikaz ...> <zarizeni ... | diskovy_oddil ...>	vypíše či nastaví vlastnosti zařízení, **-v** podrobný výpis, **--setro** nastaví práva „pouze ke čtení“, **--setrw** nastaví práva „čtení a zápisu“, **--getsize** vypíše počet sektorů na zařízení
badblocks <zarizeni | diskovy_oddil>	kontroluje na daném zařízení špatné bloky, **-b** <n> určí velikost bloků v B (implicitně 1024), **-i** <soubor> vynechá kontrolu seznamu již existujících známých špatných bloků uvedených v daném souboru, **-v** podrobný výpis `# badblocks -vb 4096 /dev/sda2 > bad_blocks` (vyhledá na daném zařízení špatné bloky a výsledek uloží do souboru)
smartctl <zarizeni>	**-a** vypíše podrobné informace o stavu daného zařízení (podporujícího SMART technologii) s cílem předejít jeho poškození, **-s** {on | off} aktivuje/deaktivuje SMART, **-t** {short | long} spustí krátký/dlouhý test `# smartctl -s on -t long /dev/sda` (aktivuje SMART a spustí dlouhý test na daném zařízení)
iotop	zobrazí interaktivním a dynamickým způsobem I/O zatížení disků běžícími procesy a rychlost zápisu a čtení dat, **-o** pouze aktivní procesy, **-u** <uzivatel> procesy daného uživatele, **-p** <PID> pouze daný proces; interaktivní volby: **q** ukončí program

ZAŘÍZENÍ A DISKOVÉ ODDÍLY	

iostat [<zarizeni ...>] [<interval> [<pocet>]]	zobrazí statistiky o využití procesoru a I/O zatížení disků, **-d** pouze zatížení disků, **-p** včetně diskových oddílů, **-x** rozšířené statistiky, **-h** v čitelném formátu
	`$ iostat -d 5`
	(vypisuje nepřetržitě informace o zatížení disků v pětisekundových intervalech)
	`$ iostat -dpxh sda sdb 1 3`
	(vypíše třikrát v sekundových intervalech podrobné informace o zatížení disků „sda“ a „sdb“ a jejich diskových oddílů)

xxd [<soubor | zarizeni>]	vypíše obsah daného souboru v hexadecimálním formátu; není-li soubor uveden, čte ze STDIN, **-l** <n> určí počet oktetů `# xxd -l 512 /dev/sda` (zobrazí obsah hlavního spouštěcího záznamu (MBR) v hexadecimálním formátu)
cat /proc/partitions	vypíše diskové oddíly, jejich hlavní (major) a vedlejší (minor) číslo a počet obsazených bloků
fdisk <zarizeni | diskovy_oddil>	vypíše počet cylindrů na daném disku; pomocí následné volby **m** zobrazí nabídku pro práci s tabulkou rozdělení disku: **p** zobrazí tabulku rozdělení disku, **n** vytvoří nový diskový oddíl, **l** vypíše možné typy oddílů, **t** nastaví typ oddílu, **d** odstraní oddíl, **q** ukončí program, **w** uloží změny na disku a ukončí program; **-l** zobrazí rozdělení daného disku, není-li specifikován, vypíše informace o rozdělení všech právě připojených disků `# fdisk -l /dev/sdb` (zobrazí tabulku rozdělení disku na daném zařízení)

ZAŘÍZENÍ A DISKOVÉ ODDÍLY	

parted [<zarizeni>] [<prikaz>]	**print** zobrazí tabulku rozdělení disku, **unit** určí jednotky pro zobrazení velikosti diskových oddílů (např. „s“, „B“, „kB“, „MB“, „MiB“, „GB“, „GiB“, „TB“, „TiB“), **mklabel** vytvoří označení disku (např. „msdos“ či „gpt“), **mkpart** <typ_oddilu | nazev_oddilu> [<typ_SS>] <zacatek> <konec> vytvoří nový diskový oddíl, **rm** <cislo_oddilu> odstraní diskový oddíl, **select** <zarizeni> vybere zařízení, které se má použít, **set** <cislo_oddilu> <priznak> <stav> nastaví stav příznaku na diskovém oddílu, **resizepart** <cislo_oddilu> <konec> změní velikost oddílu na požadovanou hodnotu, **quit** ukončí program, **-l** vypíše rozložení oddílů na všech blokových zařízeních; není-li zadán žádný příkaz, spustí se v interaktivním režimu
	`# parted /dev/vda print`
	(zobrazí tabulku rozdělení disku na „/dev/vda“)
	`# parted /dev/vda unit s print`
	(zobrazí tabulku rozdělení disku podle sektorů)
	`# parted /dev/vdb mklabel msdos`
	(zapíše MBR označení disku)
	`# parted /dev/vdb mkpart primary xfs 2048s 1001MB`
	(vytvoří primární diskový oddíl o velikosti 1 000 MB)
	`# parted /dev/vdb resizepart 1 100%`
	(rozšíří oddíl tak, aby zabíral zbývající volné místo na disku)
	`# parted /dev/vdc mklabel gpt`
	(zapíše GPT označení disku)
	`# parted /dev/vdc mkpart primary 1MiB 1001MiB; parted /dev/vdc set 1 lvm on`
	(vytvoří diskový oddíl o velikosti 1000 MiB s typem Linux LVM)
	`# parted /dev/vdc mkpart swap1 linux-swap 1001MiB 1501MiB`
	(vytvoří odkládací oddíl s názvem „swap1“)

partprobe [<zarizeni ...>]	informuje OS o změnách v tabulce rozdělení disků, **-s** vypíše souhrnné informace o blokových zařízeních a jejich oddílech
kpartx <zarizeni>	**-a** vytvoří mapovaná zařízení (v adresáři */dev/mapper*) pro diskové oddíly nalezené na daném disku, **-d** smaže mapovaná zařízení, **-u** aktualizuje mapovaná zařízení `# kpartx -a /dev/mapper/mpatha` (vytvoří device mapper zařízení pro diskové oddíly na daném disku)
dmsetup <prikaz> [<VG/LV ... | zarizeni ...>]	**ls** vypíše mapovaná zařízení (v adresáři */dev/mapper*) a jejich hlavní (major) a vedlejší (minor) čísla, **table** zobrazí mapování fyzických blokových zařízení na logická (mapovaná) zařízení – logické blokové zařízení, začátek logického sektoru, počet sektorů, typ cíle (např. „linear“, „striped“, „mirror“, „snapshot“ či „crypt“), fyzické blokové zařízení (hlavní a vedlejší čísla) a začátek fyzického sektoru, **info** zobrazí informace o mapovaném zařízení (např. jeho název, stav, hlavní a vedlejší číslo či UUID), **remove** odstraní mapované zařízení, **--target** určí typ cíle `# dmsetup ls --target crypt` (vypíše mapovaná zařízení šifrovaných blokových zařízení)

ZAŘÍZENÍ A DISKOVÉ ODDÍLY	
cryptsetup <prikaz> <diskovy_oddil | LV> [<volby> <argumenty>]	**luksFormat** zašifrujte zařízení pomocí LUKS a nastaví přístupovou frázi, **luksOpen** dešifruje šifrované zařízení a namapuje ho na dané logické device mapper zařízení, které zároveň vytvoří, **luksAddKey** přidá k šifrovanému zařízení přístupovou frázi, **luksRemoveKey** odstraní přístupovou frázi ze šifrovaného zařízení, **luksDump** vypíše informace o šifrování šifrovaného zařízení (použitou šifru, šifrovací algoritmus a klíčové sloty pro přístupovou frázi), **luksClose** odmapuje šifrované zařízení od logického device mapper zařízení `# cryptsetup luksOpen /dev/vdb1 secure` (dešifruje šifrované zařízení „/dev/vdb1“ a namapuje ho na logické device mapper zařízení „/dev/mapper/secure“)

mdadm [<rezim>] <raid_zarizeni> [<volba>] [<dilci_zarizeni>]	spravuje software RAID (virtuální) zařízení; k režimům patří **--create** vytvoří software RAID, **--grow** změní velikost či tvar aktivního pole, **--add** přidá fyzické zařízení, **--fail** označí fyzické zařízení za vadné a deaktivuje ho (vhodné k testování RAID 1 nebo RAID 5, ne však RAID 0), **--remove** odebere fyzické zařízení (nesmí být aktivní), **--detail** vypíše podrobné informace o RAID zařízení, **--stop** deaktivuje raid zařízení, **--assemble** aktivuje dříve existující raid zařízení, **--scan** aplikuje danou operaci na všechna zařízení uvedená v */etc/mdadm.conf*; **-v** podrobný výpis
	`# mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sd{b,c}1`
	(vytvoří software RAID 1 zahrnující „/dev/sdb1“ a „/dev/sdc1“)
	`# mdadm --grow /dev/md1 --raid-devices=3`
	(změní velikost raid zařízení na tři aktivní fyzická zařízení)
	`# mdadm --add /dev/md1 /dev/sdd1`
	(přidá fyzické zařízení „/dev/sdd1“ do raid zařízení „/dev/md1“)
	`# mdadm --stop /dev/md1`
	(deaktivuje raid zařízení)
	`# mdadm --assemble --scan`
	(aktivuje všechna definovaná raid zařízení)

cat /etc/mdadm.conf	vypíše informace o nastavení mdX RAID diskových polí
cat /proc/mdstat	vypíše informace o aktuálním stavu mdX RAID diskových polí
sync	synchronizuje data uložená v diskové vyrovnávací paměti s diskem (obvykle systém ukládá data s časovou prodlevou)
eject [<zarizeni>]	vysune výměnné zařízení (implicitně CD/DVD), **-t** zasune výměnné zařízení `$ eject` | `$ eject /dev/cdrom`
xinput [<volba>] [<zarizeni>]	nastaví a testuje X vstupní zařízení jako jsou myši, klávesnice a touchpady, **list** vypíše informace o zadaných vstupních zařízeních, **test** testuje zařízení, v případě chyby zobrazí nekonečnou smyčku přijatých událostí; bez argumentu vypíše seznam všech vstupních zařízení `$ xinput test 11` (testuje zařízení s ID 11)

SOUBOROVÝ SYSTÉM A SWAP	
cat /proc/filesystems	vypíše seznam souborových systémů podporovaných jádrem
cat /etc/fstab	vypíše seznam trvale definovaných souborových systémů – jméno, UUID či LABEL zařízení / diskového oddílu, přípojný bod, typ SS, volby připojení (viz příkaz „mount“), zálohování programem „dump“ (0 = ne, 1 = ano) a pořadí kontroly SS programem „fsck“ při startu systému (0 = žádná kontrola, 1 = první pořadí (pro SS „/“), 2–9 další pořadí)

mke2fs | **mkfs.ext2** | **mkfs.ext3** | **mkfs.ext4** <diskovy_oddil | LV>	vytvoří souborový systém ext2, ext3 či ext4 na daném diskovém oddílu nebo logickém svazku, **-b** <n> určí velikost bloku v B, **-c** kontroluje na zařízení špatné bloky, **-i** <n> určí počet bytů na i-node, **-t** <SS> určí typ SS (při použití „mke2fs“), **-L** <LABEL> určí název svazku SS, **-v** podrobný výpis
	`# mke2fs -t ext4 /dev/sdb2`
	(vytvoří souborový systém „ext4“ na daném diskovém oddílu)
	`# mkfs.ext4 /dev/mapper/centosvg/weblv`
	(vytvoří souborový systém „ext4“ na daném logickém svazku)

mkfs.xfs <diskovy_oddil | LV>	vytvoří souborový systém xfs na daném diskovém oddílu nebo logickém svazku, **-b size=**<n> určí velikost bloku v B (implicitně 4096 B), **-L** <LABEL> určí název svazku SS `# mkfs.xfs /dev/vg00/opt_lv` (vytvoří souborový systém „xfs“ na daném logickém svazku)
mkisofs <adresar>	vytvoří souborový systém ISO 9660 (soubor .iso) z daného adresáře, **-J** rozšíření Joliet (obsah je čitelný ve windows), **-r** rozšíření RockRidge (obsah je čitelný v Linuxu/Unixu), **-o** <soubor> určí výstupní soubor `$ mkisofs -Jro cd.iso /data` (vytvoří soubor „cd.iso“ z daného adresáře)

resize2fs <diskovy_oddil | LV> [<velikost>]	změní (zvětší či zmenší) velikost souborového systému ext2, ext3 či ext4 na požadovanou hodnotu v kB (K), MB (M), GB (G) či TB (T), SS může být při zvětšování připojen; není-li velikost SS při jeho zvětšování uvedena, automaticky vyplní daný oddíl až do jeho max. velikosti (pokud SS zaplňuje celý oddíl, který je třeba rozšířit, musí se nejdřív zvětšit daný oddíl a pak teprve SS; je-li nutné oddíl zmenšit, musí se nejdřív odpojit, zkontrolovat SS, snížit velikost SS a až poté zmenšit velikost oddílu tak, aby nebyl menší než nová velikost SS), **-p** vypíše průběh v procentech
	`# resize2fs /dev/sdb3 500M`
	(nastaví velikost souborového systému na 500 MB)
	`# resize2fs /dev/vg00/homelv`
	(nastaví velikost souborového systému podle velikosti logického svazku)

xfs_growfs <SS | diskovy_oddil | LV>	zvětší velikost souborového systému xfs (SS musí být připojen), **-D** <velikost> určí velikost SS (není-li velikost SS uvedena, automaticky vyplní daný oddíl až do jeho max. velikosti)
dumpe2fs <SS | diskovy_oddil | LV>	zobrazí informace o souborovém systému ext2, ext3 či ext4
xfs_info <SS | diskovy_oddil | LV>	zobrazí informace o souborovém systému xfs

SOUBOROVÝ SYSTÉM A SWAP	

tune2fs <diskovy_oddil | LV>	změní parametry souborového systému ext2, ext3 či ext4, **-l** vypíše obsah superbloku, **-L** <LABEL> určí název svazku SS, **-j** přidá žurnál ext3 do SS ext2 (ten musí být nejdřív odpojen; pro trvalé nastavení je třeba upravit i údaj „ext2“ na „ext3“ v */etc/fstab*), **-O** (^)<vlastnost> nastaví či při použití znaku „^“ odstraní danou vlastnost SS, **-o** (^)<volba_pripojeni> nastaví či při použití znaku „^“ odstraní danou volbu připojení SS
	`# tune2fs -j /dev/sdb1`
	(změní typ souborového systému „ext2“ na žurnálový „ext3“)
	`# tune2fs -O ^has_journal /dev/sdb1`
	(změní typ souborového systému „ext3“ zpět na „ext2“)
	`# tune2fs -o acl /dev/sdb1`
	(povolí ACL na souborovém systému)

xfs_admin <diskovy_oddil | LV>	změní parametry souborového systému xfs (SS musí být odpojen, „/“ v režimu „pouze ke čtení“), **-L** <LABEL> určí název svazku SS
debugfs [<diskovy_oddil | LV>]	interaktivně kontroluje a případně upravuje souborový systém ext2, ext3 či ext4 (SS musí být odpojen, „/“ v režimu „pouze ke čtení“), **-w** zpřístupní SS v režimu „čtení a zápisu“; **open** <diskovy_oddil> zpřístupní SS ke kontrole či editaci, **close** opustí SS, **icheck** <blok> vypíše číslo i-uzlu využívajícího daný blok, **ncheck** <i-uzel> vypíše cestu k danému i-uzlu (souboru), **quit** ukončí program
xfs_db <diskovy_oddil | LV>	interaktivně kontroluje a případně upravuje souborový systém xfs (SS musí být odpojen, „/“ v režimu „pouze ke čtení“), **quit** ukončí program

e2fsck | **fsck.ext2** | **fsck.ext3** | **fsck.ext4** <diskovy_oddil | LV>	kontroluje a případně opraví souborový systém ext2, ext3 či ext4 (SS musí být odpojen, „/“ v režimu „pouze ke čtení“), **-b** <superblock> použije daný superblock, **-c** kontroluje na zařízení špatné bloky, **-f** provede kontrolu, i když se SS zdá být čistý, **-n** kontroluje SS neinteraktivní formou a odpoví na všechny otázky „ne“, **-p** automaticky opraví SS, **-v** podrobný výpis
	`# e2fsck -n /dev/sdb1`
	(zkontroluje souborový systém)
	`# e2fsck /dev/sdb1 -b 97608`
	(opraví souborový systém za použití alternativního superbloku)

xfs_repair <diskovy_oddil | LV>	kontroluje a opraví souborový systém xfs (SS musí být odpojen, „/“ v režimu „pouze ke čtení“), **-n** provede kontrolu bez úpravy SS, **-v** podrobný výpis
	`# xfs_repair -n /dev/sdb1`
	(zkontroluje souborový systém)
	`# xfs_repair /dev/sdb1`
	(zkontroluje a automaticky opraví souborový systém)

SOUBOROVÝ SYSTÉM A SWAP	

dump [<zdroj>]	zálohuje souborový systém ext2, ext3 či ext4 (SS musí být odpojen, „/“ v režimu „pouze ke čtení“), **-**<n> určí úroveň inkrementální zálohy (nabývá hodnot 0–9, přičemž 0 = plná záloha), **-a** automatická velikost (zapisuje data do konce pásky), **-f** <cil> určí cílový soubor či zařízení, **-z** komprimace gzip, **-j** komprimace bzip2, **-u** aktualizuje */etc/dumpdates* (přehled provedených záloh), **-v** podrobný výpis, **-L** <nazev> pojmenuje archiv, **-M** umožní „multi-volume“ zálohu (při archivech větších než 2 GB), **-S** určí velikost místa potřebného k provedení zálohy, **-W** vypíše SS, jenž je třeba zálohovat (údaje z */etc/dumpdates* a */etc/fstab*)
	`# dump -0uf /tmp/home.bckp /home`
	(zálohuje domovský adresář do daného souboru)
	`# dump -0auf /dev/st0 -L Full_bckp_11-07-08 /`
	(zálohuje celý systém na pásku pod daným názvem)

xfsdump [<zdroj>]	zálohuje souborový systém xfs (SS musí být odpojen, „/“ v režimu „pouze ke čtení“), **-l** <n> určí úroveň inkrementální zálohy (nabývá hodnot 0–9, přičemž 0 = plná záloha), **-f** <cil> určí cílový soubor či zařízení, **-v** podrobný výpis, **-I** zobrazí xfsdump inventář */var/lib/xfsdump/inventory/** (přehled provedených záloh), **-L** <nazev> pojmenuje archiv `# xfsdump -f /dev/st0 /` (zálohuje celý systém na pásku)

restore	obnoví soubory nebo souborové systémy ze záloh vytvořených programem „dump“ do pracovního adresáře, **-f** <zdroj> určí zdrojový soubor či zařízení, **-r** nově naformátovaný SS, **-x** existující SS, **-s** <n> číslo zálohy (souboru), **-i** interaktivní režim, **-t** vypíše obsah archivu, **-v** podrobný výpis
	`# restore -rvf /dev/rmt0`
	(provede úplné obnovení souborového systému z pásky)
	`# restore -is 3 -f /dev/rmt0`
	(provede interaktivní obnovu dat – je možno obnovit i jednotlivé soubory)

xfsrestore [<cil>]	obnoví soubory nebo souborové systémy ze záloh vytvořených programem „xfsdump“, **-f** <zdroj> určí zdrojový soubor či zařízení, **-i** interaktivní režim, **-r** provede kumulativní obnovu z plných a inkrementálních záloh, **-t** vypíše obsah archivu, **-v** podrobný výpis, **-L** <nazev> určí název archivu, který má být obnoven, **-R** pokračuje ve dříve přerušené obnově `# xfsrestore -f /dev/st0 /` (provede úplné obnovení souborového systému z pásky)
quota	vypíše informace o využití diskového prostoru a nastavení limitů (kvót) v kB pro přihlášeného uživatele na všech připojených souborových systémech ext2, ext3 či ext4, **-f** <SS> pro daný souborový systém, **-u** <uzivatel> pro daného uživatele, **-g** <skupina> pro danou skupinu, **-s** v čitelném formátu, **-v** podrobný výpis

SOUBOROVÝ SYSTÉM A SWAP	

xfs_quota	běží v základním režimu a interaktivně vypíše informace o využití diskového prostoru a nastavení limitů (kvót) na všech připojených souborových systémech xfs, **-x** běží v expertním režimu a interaktivně vypíše nebo nastaví kvóty na souborových systémech xfs, **-c** <prikaz> spustí dílčí příkaz přímo z příkazového řádku (neinteraktivní režim)
	`# xfs_quota -xc 'report -h' /home`
	(zobrazí informace o kvótách pro souborový systém „/home“)
	`# xfs_quota -xc 'limit bsoft=1000m bhard=1500m marek' /home`
	(nastaví měkký a tvrdý limit 1000 MB a 1500 MB pro uživatele „marek“ na souborovém systému „/home“)

repquota [<SS ...>]	vypíše souhrnné informace o využití a nastavení kvót na daném SS, **-a** na všech připojených SS, **-u** kvóty uživatelů (implicitně), **-g** kvóty skupin, **-s** v čitelném formátu, **-v** podrobný výpis

quotacheck [<SS>]	kontroluje nastavení kvót (max. velikost použitelného prostoru v i-nodech nebo v 1kB blocích pro uživatele či skupiny) na daném SS s povoleným atributem „usrquota“ či „grpquota“ v */etc/fstab* (před kontrolou připojí daný SS v režimu „pouze ke čtení“, po ní opět v režimu „čtení a zápisu“), **-a** kontroluje všechny připojené SS mimo NFS, **-c** provede novou kontrolu a vytvoří soubory *aquota.user* či *aquota.group* (ignoruje existující kvóty), **-u** kvóty uživatelů (implicitně), **-g** kvóty skupin, **-v** podrobný výpis, **-i** interaktivní režim
	`# quotacheck -cvug /home`
	(aktivuje použití kvót na daném SS)
	`# quotacheck -avug`
	(vypíše využití všech připojených SS s povolenými kvótami)

edquota	nastaví kvóty editací souboru *aquota.user* či *aquota.group*, **-u** <uzivatel> danému uživateli, **-g** <skupina> dané skupině, **-p** <vzor> podle daného vzoru (uživatele, skupiny či projektu), **-f** <SS> pro daný SS, **-t** nastaví max. období (v řádu sekund, minut, hodin či dní), během něhož může být překročen „soft limit“ až do hodnoty uvedené jako „hard limit“ (není-li tato doba nastavena, max. hodnota = „soft limit“) `# edquota -p rob $(awk -F: '$3 > 499 {print $1}' /etc/passwd)` (nastaví kvóty všem uživatelům, jejichž UID je větší než 499 podle uživatele „rob“)

SOUBOROVÝ SYSTÉM A SWAP	
setquota [<SS ...>]	nastaví kvóty přímo z příkazového řádku, **-u** <uzivatel> danému uživateli, **-g** <skupina> dané skupině, **-p** <vzor> podle daného vzoru (uživatele, skupiny či projektu), **-a** pro všechny připojené SS s nastaveným parametrem „usrquota“/„grpquota“, **-t** nastaví max. období (v řádu sekund, minut, hodin či dní), během něhož může být překročen „soft limit“ až do hodnoty uvedené jako „hard limit“ (není-li tato doba nastavena, max. hodnota = „soft limit“) `# setquota -u tim 512 1024 0 0 /home` (nastaví uživateli „tim“ kvóty na daný SS v pořadí „block-softlimit block-hardlimit inode-softlimit inode-hardlimit“)
quotaon [<SS ...>]	povolí použití kvót na daném SS, **-a** na všech připojených SS, **-u** kvóty uživatelů (implicitně), **-g** kvóty skupin
quotaoff [<SS ...>]	zakáže použití kvót na daném SS, **-a** na všech připojených SS, **-u** kvóty uživatelů (implicitně), **-g** kvóty skupin
cat /etc/mtab | **cat /proc/mounts** | **cat /proc/self/mounts**	vypíše seznam právě připojených souborových systémů

SOUBOROVÝ SYSTÉM A SWAP	

mount [<zarizeni | diskovy_oddil | LV | adresar | NFS_server:SS [adresar]>]	vypíše právě připojené souborové systémy (údaje z */etc/mtab*) či připojí daný SS, **-a** připojí všechny SS v */etc/fstab* s volbou „defaults“ nebo „auto“, **-L** <LABEL> podle názvu svazku, **-U** <UUID> podle UUID, **-t** <SS> určí typ SS, **-n** bez zápisu do */etc/mtab* (např. má-li být SS /etc připojen v režimu „pouze ke čtení“), **-o** <atribut> určí volby připojení oddělené čárkou (např. **remount** znovu připojí již připojený SS – **ro** „pouze ke čtení“, **rw** „čtení a zápis“, (**no**)**dev** (ne)povolí přístup k SS znakovým a blokovým zařízením, (**a**)**sync** (ne)povolí synchronní čtení a zápis dat na SS, (**no**)**auto** (ne)připojí SS automaticky, **users** povolí připojení i odpojení SS jakémukoliv uživateli, (**no**)**user** (ne)povolí připojení SS běžnému uživateli (ale jen ten ho smí opět odpojit), **usrquota**/**grpquota** povolí využití disku a limity na SS pro jednotlivé uživatele/skupiny (atribut musí existovat v */etc/fstab*), (**no**)**acl** (ne)povolí rozšířená přístupová práva ACL pro SS ext2, ext3 či ext4, (**no**)**atime** (ne)povolí úpravu údajů o posledním přístupu k SS v jeho i-nodu, (**no**)**suid** (ne)povolí práva setuid a setgid spustitelným souborům, (**no**)**exec** (ne)povolí spuštění binárních souborů a skriptů na SS, **defaults** = rw, suid, dev, exec, auto, nouser, async)
	`# mount /dev/fd0` | `$ mount /mnt/floppy`
	(připojí disketu – údaje pro připojení zařízení jsou uvedeny v */etc/fstab*)
	`# mount -t iso9660 /dev/cdrom /mnt/cdrom`
	(připojí obsah CD/DVD do daného adresáře)
	`# mount -t iso9660 -o loop /tmp/image.iso /mnt/iso`
	(připojí obsah ISO souboru do daného adresáře)
	`# mount -t vfat /dev/sdc1 /mnt/flash`
	(připojí obsah USB flash disku do daného adresáře)
	`# mount -no remount,ro /`
	(připojí kořenový souborový systém v režimu „pouze ke čtení“)
	`# mount server_01:/data /mnt/nfs`
	(připojí „/data“ ze vzdáleného serveru do lokálního adresáře přes NFS)
	`# mount LABEL="Test Label" /test`
	(připojí souborový systém do daného adresáře podle názvu svazku)
	`# mount UUID="b3cb79b8-a8c6-f188-064b-82236deb1604" /test`
	(připojí souborový systém do daného adresáře podle svého UUID)

umount <zarizeni ... | diskovy_oddil ... | LV ... | adresar ...>	odpojí souborový systém (nesmí být používán žádným procesem), **-a** všechny připojené SS, **-f** vynutí odpojení SS `$ umount /mnt/flash` | `$ umount /dev/sdc1` (odpojí USB flash disk připojený do daného adresáře či přiřazený k danému zařízení)
mountpoint <adresar>	zjistí, zda je daný adresář přípojným bodem (údaje z */proc/self/mountinfo*), **-d** vypíše hlavní a vedlejší číslo zařízení, které je připojeno do adresáře

SOUBOROVÝ SYSTÉM A SWAP	
showmount [<pocitac>]	vypíše seznam všech počítačů (klientů), které mají povoleno připojení lokálních adresářů přes NFS z lokálního či daného serveru (údaje z */etc/exports(.d/*)*), **-a** vypíše názvy či IP adresy klientů a adresáře, které jsou jim právě exportovány, **-e** vypíše seznam všech exportovaných adresářů a klientů, jenž mají povoleno je připojit

exportfs [<volba>] [<klient>**:**<adresar>]	definuje vzdálené počítače (klienty) a lokální adresáře, které jim budou exportovány přes NFS (trvalé nastavení se provede v */etc/exports(.d/*)*); bez argumentu či s volbou **-s** vypíše seznam všech exportovaných adresářů, **-a** exportuje všechny adresáře uvedené v */etc/exports(.d/*)*, **-r** provede opakovaný export položek z */etc/exports(.d/*)* (synchronizuje soubor */var/lib/nfs/etab* s údaji v */etc/exports(.d/*)*), **-i** ignoruje informace v */etc/exports(.d/*)* a použije pouze volby zadané přímo v příkazovém řádku, **-u** <klient>**:**<adresar> zruší daný export, s volbou „-a“ zruší export všech položek v */etc/exports(.d/*)*, **-o** určí další volby pro export lokálního adresáře (výchozí volby jsou **sync,ro,root_squash,wdelay**), **-v** podrobný výpis
	`# exportfs -av`
	(exportuje všechny adresáře uvedené v */etc/exports(.d/*)*)
	`# exportfs -o rw client1:/usr/tmp`
	(povolí vzdálenému počítači zápis do daného adresáře)
	`# exportfs -au`
	(zruší export všech adresářů uvedených v */etc/exports(.d/*)*)

nfsstat	zobrazí statistiky o aktivitách NFS klienta a serveru, **-c** pouze statistiky NFS klienta, **-s** pouze statistiky NFS serveru, **-m** zobrazí podrobné informace o připojených NFS souborových systémech
nfsiostat [<pripojny_bod ...>] [<interval> [<pocet>]]	zobrazí I/O statistiky NFS klienta na daných či všech NFS přípojných bodech (údaje z */proc/self/mountstats*) `$ nfsiostat /mnt/nfs 10 5` (zobrazí pět I/O statistik na daném přípojném bodu v desetisekundových intervalech)

df [<soubor ... | adresar ... | diskovy_oddil ... | LV ...>]	vypíše informace o využití diskového prostoru na daném či všech právě připojených souborových systémech v kB (název zařízení souborového systému, celkovou velikost, využitý prostor, dostupný prostor, procento využitého prostoru a přípojný bod), **-m** v MB, **-i** využití kapacity v i-uzlech, **-h** v čitelném formátu, **-l** pouze lokální SS, **-T** zobrazí typ všech SS, **-t** <SS> pouze SS daného typu, **-P** použije výstupní formát POSIX
	`$ df -ml | sed '1 d' | awk '{sum+=$3} END {printf "%.f\n",sum}'`
	(vypíše celkový využitý prostor v MB, který zabírají všechny lokální souborové systémy)
	`$ df -P | awk '+$5 >= 90 {print}'`
	(vypíše souborové systémy s využitou kapacitou 90 % a více)

mkswap <diskovy_oddil | LV>	vytvoří odkládací prostor (swap) pro virtuální paměť, **-L** <LABEL> určí název svazku

SOUBOROVÝ SYSTÉM A SWAP	
swapon <diskovy_oddil ... | LV ...>	aktivuje odkládací prostor (pro trvalou aktivaci musí být oddíl uveden v */etc/fstab* s volbou „auto“), **-a** aktivuje všechny swap oddíly v */etc/fstab*, **-s** vypíše aktivní oddíly s odkládacím prostorem (údaje z */proc/swaps*), **-v** podrobný výpis (název diskového oddílu, velikost, využití a priorita)
swapoff <diskovy_oddil ... | LV ...>	deaktivuje odkládací prostor, **-a** deaktivuje všechny swap oddíly v */etc/fstab*, **-v** podrobný výpis
cat /proc/swaps	vypíše aktivní oddíly s odkládacím prostorem
photorec	obnoví smazaná data z HDD či přenosného média různých typů souborových systémů; spustí se v interaktivním režimu a automaticky rozpozná SS; lze obnovit i konkrétní datový formát

LOGICAL VOLUME MANAGER (LVM)	**Virtualizace a dynamická správa diskových oddílů**
Physical Volume (PV)	**Fyzické blokové zařízení nebo diskový oddíl (fyzický svazek)**
pvcreate <zarizeni ... | diskovy_oddil ...>	vytvoří fyzický svazek (PV) z daných disků či diskových oddílů `# pvcreate /dev/sda2 /dev/sdb` (vytvoří fyzický svazek z daného diskového oddílu a disku)
pvs	vypíše všechny fyzické svazky (PV) v systému, jejich skupinu svazků (VG), atributy, velikost a volné místo
pvdisplay [<PV ...>]	vypíše podrobné informace o všech či daných fyzických svazcích (PV), **-m** zobrazí mapování fyzických bloků (PE) na logické svazky (LV) a logické bloky (LE)
pvchange <PV ...>	změní atributy fyzického svazku (PV), **-x** {y | n} povolí/zakáže alokaci fyzických bloků (PE) na fyzickém svazku (PV)

pvresize <PV ...>	změní velikost fyzického svazku (PV)
	`# pvresize --setphysicalvolumesize 40G /dev/sda1`
	(nastaví velikost fyzického svazku)
	`# pvresize /dev/sda4`
	(rozšíří fyzický svazek o všechno volné místo na /dev/sda4)

pvmove <zdrojovy_PV> [<cilovy_PV>]	přesune data z jednoho fyzického svazku (PV) na druhý, není-li cílový fyzický svazek (PV) určen, data se přemístí na jakékoliv volné místo v rámci dané skupiny svazků (VG)
pvremove <PV ...>	odstraní fyzický svazek (PV) z LVM
Volume Group (VG)	**Úložný prostor z jednoho nebo více fyzických svazků (skupina svazků)**
vgcreate <VG> <PV ...>	vytvoří skupinu svazků (VG) z daných fyzických svazků (PV), **-s** <velikost> určí velikost fyzických bloků (PE) na fyzických svazcích (PV) v této skupině svazků (VG) (standardně 4 MB) `# vgcreate vg00 /dev/sda2 /dev/sdb` (vytvoří skupinu svazků „vg00“ z daných fyzických svazků)
vgs	vypíše všechny skupiny svazků (VG) v systému, počet jejich fyzických svazků (PV), logických svazků (LV) a kopií, atributy, velikost a volné místo

LOGICAL VOLUME MANAGER (LVM)	**Virtualizace a dynamická správa diskových oddílů**
vgdisplay [<VG ...>]	vypíše podrobné informace o všech či daných skupinách svazků (VG), **-v** včetně jejich fyzických svazků (PV) a logických svazků (LV)
vgchange <VG ...>	změní atributy skupiny svazků (VG), **-a** {y | n} aktivuje/deaktivuje skupinu svazků (VG), **-l** <n> nastaví maximální počet logických svazků (LV) ve skupině svazků (VG), **-p** <n> nastaví maximální počet fyzických svazků (PV) ve skupině svazků (VG), **-x** {y | n} povolí/zakáže přidání nebo odebrání fyzických svazků (PV) do/ze skupiny svazků (VG) `# vgchange -a n data_vg` (deaktivuje skupinu svazků)
vgrename <stara_VG> <nova_VG>	přejmenuje skupinu svazků (VG) `# vgrename data_vg datavg` (přejmenuje skupinu svazků „data_vg“ na „datavg“)
vgextend <VG> <PV ...>	přidá fyzický svazek (PV) do skupiny svazků (VG) `# vgextend vg00 /dev/sda4` (přidá fyzický svazek „/dev/sda4“ do skupiny svazků „vg00“)
vgreduce <VG> <PV ...>	odstraní fyzický svazek (PV) ze skupiny svazků (VG) (fyzický svazek (PV) musí být prázdný)
vgexport <VG ...>	exportuje skupinu svazků (VG) (znepřístupní ji systému)
vgimport <VG ...>	importuje skupinu svazků (VG) (zpřístupní exportovanou skupinu svazků (VG) systému)
vgsplit <zdrojova_VG> <cilova_VG> [<PV ...>]	přesune fyzický svazek (PV) do nové nebo existující skupiny svazků (VG), **-n** <LV> přesune pouze fyzické svazky (PV) používané daným logickým svazkem (LV), **-l** <n> nastaví maximální počet logických svazků (LV) v dané skupině svazků (VG), **-p** <n> nastaví maximální počet fyzických svazků (PV) v dané skupině svazků (VG) `# vgsplit vg01 vg02 /dev/sde1` (přesune fyzický svazek „/dev/sde1“ ze skupiny svazků „vg01“ do „vg02“)
vgmerge <cilova_VG> <zdrojova_VG>	sloučí neaktivní zdrojovou skupinu svazků (VG) do cílové skupiny svazků (VG) (velikost fyzických bloků (PE) musí být stejná a součty fyzických svazků (PV) a logických svazků (LV) obou skupin svazků (VG) se musí vejít do limitů cílové skupiny svazků (VG)) `# vgmerge vg01 vg02` (sloučí „vg02“ do „vg01“, přičemž zachová všechny vlastnosti skupiny svazků „vg01“)
vgremove <VG ...>	odstraní skupinu svazků (VG)
vgcfgrestore <VG>	obnoví metadata skupiny svazků (VG) z poslední zálohy, **-l** vypíše seznam všech archivovaných metadat (archivních souborů) pro danou skupinu svazků (VG), **-f** <archivni_soubor> obnoví konfiguraci skupiny svazků (VG) z daného souboru `# vgcfgrestore -f /etc/lvm/archive/vg00_00031-1442492648.vg vg00` (obnoví konfiguraci skupiny svazků „vg00“ z daného souboru)

LOGICAL VOLUME MANAGER (LVM)	**Virtualizace a dynamická správa diskových oddílů**
Logical Volume (LV)	**Virtuální diskový oddíl (logický svazek) v rámci skupiny svazků**

lvcreate <LV> <VG>	vytvoří logický svazek (LV) v dané skupině svazků (VG), **-n** <LV> určí název logického svazku (LV), **-L** <velikost> určí velikost logického svazku (LV), **-l** <n> určí velikost logického svazku (LV) zadáním počtu logických bloků (LE), **-m** <n> určí počet kopií logického svazku (LV) (vytvoří zrcadlení lineárního logického svazku (LV)), **-p** {r | rw} nastaví práva pro čtení nebo čtení a zápis
	`# lvcreate -n web_lv -L 10G datavg`
	(vytvoří logický svazek „web_lv“ o velikosti 10 GB ve skupině svazků „datavg“)
	`# lvcreate -n db_lv -l 25%VG datavg`
	(vytvoří logický svazek „db_lv“, který zabírá 25 % skupiny svazků „datavg“)

lvs	vypíše všechny logické svazky (LV) v systému, jejich skupinu svazků (VG), atributy, velikost a další informace `# lvs -a -o segtype,devices,lv_name,vg_name` (vypíše typ logického svazku (linear, striped, mirrored, snapshot), fyzický svazek, název logického svazku a skupinu svazků)
lvdisplay [<VG/LV ... | VG ...>]	vypíše podrobné informace o všech či daných logických svazcích (LV) nebo všech logických svazcích (LV) v dané skupině svazků (VG), **-m** zobrazí mapování logických bloků (LE) na fyzické svazky (PV) a fyzické bloky (PE) a typ logického svazku (LV) (linear, striped, mirrored, snapshot)
lvchange <VG/LV ...>	změní atributy logického svazku (LV), **-a** {y | n} aktivuje/deaktivuje logický svazek (LV), **-p** {r | rw} nastaví práva pro čtení nebo čtení a zápis `# lvchange -a n datavg/web_lv` (deaktivuje logický svazek)
lvrename <stary_VG/LV> <novy_VG/LV>	přejmenuje logický svazek (LV) `# lvrename datavg/web_lv datavg/weblv` (přejmenuje logický svazek „web_lv“ na „weblv“)

lvresize <VG/LV ...>	změní velikost logického svazku (LV), **-L** [**+** | **-**]<velikost> nastaví, zvětší či zmenší velikost logického svazku (LV) na požadovanou hodnotu v MB (M), GB (G), TB (T), PB (P), EB (E), **-l** [**+** | **-**]<n> určí velikost logického svazku (LV) zadáním počtu logických bloků (LE) (pokud souborový systém zaplňuje celý logický svazek (LV), který je třeba rozšířit, musí se nejdřív zvětšit daný logický svazek (LV) a pak teprve souborový systém; je-li nutné logický svazek (LV) zmenšit, musí se nejdřív odpojit, zkontrolovat souborový systém, snížit velikost souborového systému a až poté zmenšit velikost logického svazku (LV) tak, aby nebyl menší než nová velikost souborového systému), **-r** změní velikost odpovídajícího souborového systému
	`# lvresize -rL +200M vg00/rootlv`
	(zvětší velikost logického svazku včetně souborového systému o 200 MB)
	`# lvresize -l +100%FREE /dev/VolGroup00/LogVol00`
	(rozšíří logický svazek o zbývající volné místo ve skupině svazků)

LOGICAL VOLUME MANAGER (LVM)	**Virtualizace a dynamická správa diskových oddílů**
lvconvert <VG/LV>	změní typ logického svazku (LV), **-m** <n> určí počet kopií logického svazku (LV) (vytvoří zrcadlení lineárního logického svazku (LV)), **-s** <zdrojovy_LV> <cilovy_LV> vytvoří zálohu (snapshot) logického svazku (LV) `# lvconvert -s vg00/lvol1 vg00/lvol2` (vytvoří zálohu (snapshot) logického svazku)
lvremove <VG/LV ... | VG ...>	odstraní logický svazek (LV) nebo všechny logické svazky (LV) v dané skupině svazků (VG)

POKROČILÉ FUNKCE ÚLOŽIŠTĚ	
STRATIS (platí od RHEL 8)	**spravuje disková pole fyzických úložných zařízení a v případě potřeby rozšiřuje souborové systémy**
stratis pool create <pole> <zarizeni ...>	vytvoří diskové pole z jednoho nebo více blokových zařízení `# stratis pool create pool1 /dev/vdb` (vytvoří diskové pole „pool1“ z daného disku)
stratis pool list	zobrazí všechna disková pole v systému
stratis pool rename <stare_pole> <nove_pole>	přejmenuje diskové pole
stratis pool destroy <pole ...>	odstraní diskové pole
stratis pool add-data <pole> <zarizeni ...>	přidá blokové zařízení do diskového pole `# stratis pool add-data pool1 /dev/vdc` (přidá disk do diskového pole „pool1“)
stratis blockdev list [<pole>]	zobrazí bloková zařízení v diskovém poli; bez argumentu zobrazí bloková zařízení ve všech diskových polích
stratis fs create <pole> <SS ...>	vytvoří dynamický a flexibilní souborový systém v diskovém poli (v */stratis/<pole>/<FS>*) `# stratis fs create pool1 fs1` (vytvoří souborový systém „fs1“ v diskovém poli „pool1“)
stratis fs snapshot <pole> <SS> <snapshot>	vytvoří snapshot daného souborového systému v diskovém poli (v */stratis/<pole>/<FS_snap>*) `# stratis fs snapshot pool1 fs1 fs1_snap` (vytvoří snapshot „fs1_snap“ souborového systému „fs1“ v diskovém poli „pool1“)
stratis fs list [<pole>]	zobrazí souborové systémy a jejich snapshoty v daném diskovém poli; bez argumentu zobrazí souborové systémy a jejich snapshoty ve všech diskových polích
stratis fs rename <pole> <stary_SS> <novy_SS>	přejmenuje souborový systém
stratis fs destroy <pole> <SS ... | snapshot ...>	odstraní souborový systém nebo snapshot z daného diskového pole `# stratis fs destroy pool1 fs1`|`# stratis fs destroy pool1 fs1_snap` (odstraní souborový systém „fs1“ / snapshot „fs1_snap“ z diskového pole „pool1“)
VDO (Virtual Data Optimizer) (platí od RHEL 8)	**snižuje spotřebu úložného prostoru na blokových zařízeních a minimalizuje replikaci dat**

POKROČILÉ FUNKCE ÚLOŽIŠTĚ	
vdo create --name=<svazek> **--device=**<zarizeni> **--vdoLogicalSize=**<velikost>	vytvoří a spustí VDO svazek z daného zařízení a o dané velikosti `# vdo create --name=vdo1 --device=/dev/vdd --vdoLogicalSize=500G` (vytvoří VDO svazek „vdo1“ z disku „/dev/vdd“ o velikosti 500 GB)
vdo list	zobrazí všechny spuštěné VDO svazky
vdo start --name=<svazek>	spustí zastavený, aktivovaný VDO svazek, **--all** všechny zastavené a aktivované VDO svazky
vdo stop --name=<svazek>	zastaví spuštěný VDO svazek, **--all** všechny spuštěné VDO svazky
vdo activate --name=<svazek>	aktivuje daný VDO svazek, **--all** všechny VDO svazky
vdo deactivate --name=<svazek>	deaktivuje daný VDO svazek, **--all** všechny VDO svazky
vdo status --name=<svazek>	zobrazí nastavení pro daný VDO svazek; není-li uveden, zobrazí nastavení pro všechny VDO svazky
vdostats	zobrazí využití všech spuštěných VDO svazků, **--human-readable** v čitelném formátu
vdo remove --name=<svazek>	odstraní VDO svazek, **--all** všechny VDO svazky

From:
https://prompt.cz/ - **Prompt.cz**

Permanent link:
https://prompt.cz/zarizeni-a-souborovy-system

Last update: **2024/07/06 00:47**