



**Algebra**  
visoko učilište

# Napredni datotečni sustavi

[www.racunarstvo.hr](http://www.racunarstvo.hr)

# Teme

- RAMdisk
- LVM
- ZFS
- XFS
- iSCSI



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# RAMdisk

- ideja - rezervirati dio RAM memorije, na tom dijelu RAM memorije napraviti "particiju", formatirati ju sa nekim podržanim datotečnim sustavom i koristiti kao uobičajeni direktorij na sustavu
- dva načina kako napraviti RAMdisk na Linuxu - korištenjem *ramfs* i *tmpfs* datotečnin sustava



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# ramfs

- ramfs – zastarjelo, rezerviranje fizičke memorije direktnim formatiranjem

- primjer:

```
# mkfs.ext3 /dev/ram0 16384
```

- nakon rezervacije i formatiranja, procedura je ista kao i kod svih drugih datotečnih sustava – napraviti direktorij za *mount* i montirati
- problem – ne postoji trajna konfiguracija, nakon restarta nestaju svi podaci



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# tmpfs

- ponaša se kao klasični datotečni sustav neovisno o tome gdje se nalazi, pošto može koristiti i RAM memoriju i swap particiju
- konfiguracija *tmpfs*-a se već nekoliko godina radi automatski pri instalaciji Linuxa, pošto je to primjena koncepta zajedničke memorije (*shm*, *Shared Memory*) na uobičajene postupke poput pohrane datoteka i direktorija na disk/particiju koja se zapravo nalazi u memoriji
- primjer iz `/etc/fstab`:

```
tmpfs    /dev/shm tmpfs defaults 0 0
```

Mogli bismo i promijeniti konfiguraciju navedenog datotečnog sustava tako da eksplicitno zadamo veličinu:

```
tmpfs    /dev/shm tmpfs defaults,size=2G 0 0
```



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# LVM (Logical Volume Manager)

- jedan od najčešćih problema sa sustavom za pohranu - nemogućnost kvalitetnog upravljanja diskovnim prostorom
- Pitanje:

Kako povećati ili smanjiti postojeće particije i datotečne sustave kako bismo slobodni prostor npr. dodijelili nekoj drugoj particiji kojoj u tom trenutku prostor fali?



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# Faze kreiranja LVM-a

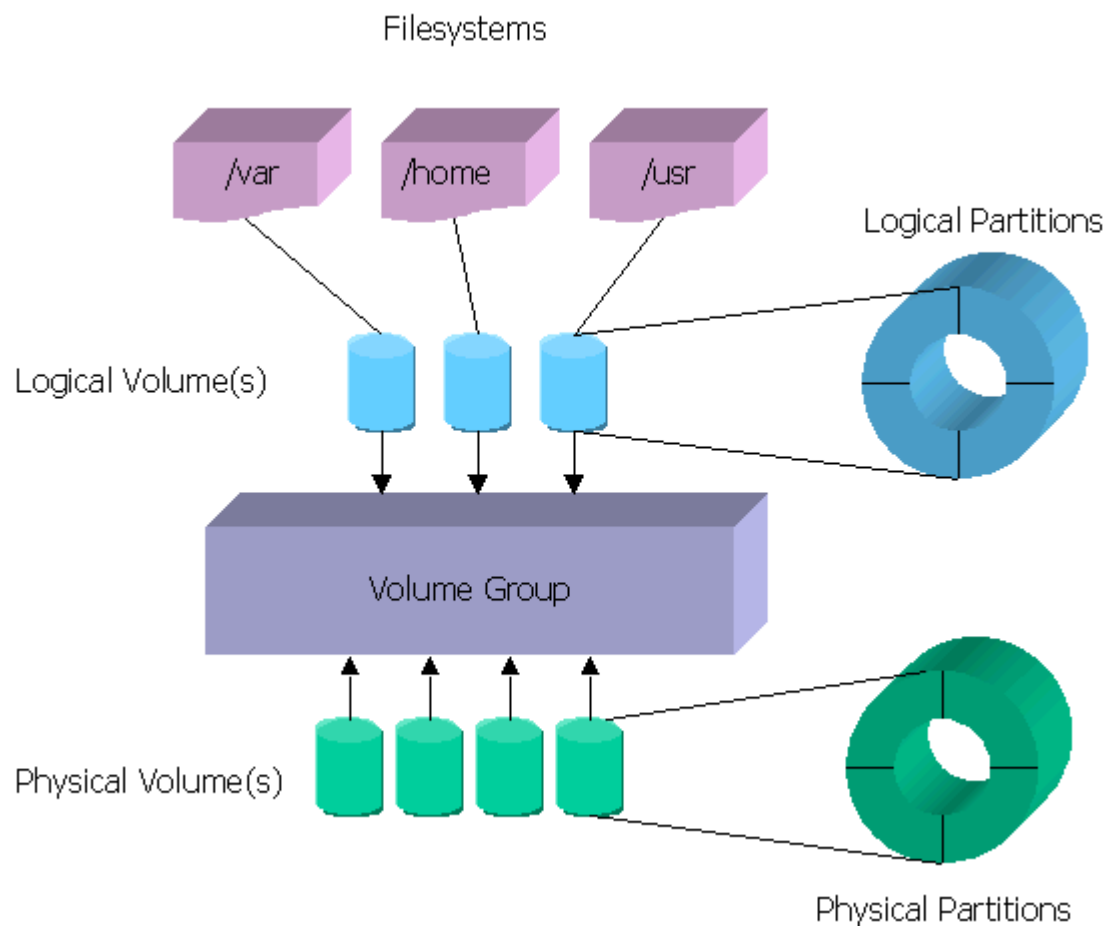
1. Izrada particije za LVM, tj. particioniranje (iako se mogu koristiti i cijeli diskovi);
2. Formatiranje particija koje smo pripremili za LVM, tj. postavljanje "LVM datotečnog sustava" korištenjem komande *pvcreate*;
3. Kreiranje grupe kapaciteta (*Volume Group*), koja se koristi radi *okrupnjavanja* dostupnih kapaciteta;
4. Kreiranje logičkog volume-a (*Logical Volume*)



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# Faze kreiranja LVM-a



**Algebra**

visoka škola  
za primijenjeno  
računarstvo



# Physical Extent

- *Physical Extent* (PE) - LVM-ov *cluster size* - minimalna količina kapaciteta koju možemo koristiti da bismo iz grupe kapaciteta napravili logički volume.
- Primjer: ako je veličina PE 4MB a mi pokušavamo napraviti logički volume veličine 250MB, pri kreiranju logičkog volume-a će biti odabran najbliži kapacitet traženom (252MB), koji je *djeljiv* sa 4MB (250 ne možemo podijeliti sa 4 i dobiti cijeli broj).
- Način razmišljanja u praksi: ako radimo LVM zato da bismo na njega smjestili datoteke i direktorije od korisnika za SAMBA *file share*, onda će manji PE biti dobar izbor. Ako pak želimo na LVM pohraniti datoteke od baze podataka (velike datoteke!), veći PE će bolje odgovarati toj specifičnoj primjeni.



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# *Grow i shrink*

- Povećavanje LVM-a (*grow*) ide u dvije faze:
  1. Povećavanje logičkog volumea, maksimalno za veličinu preostalog slobodnog prostora unutar grupe kapaciteta;
  2. Povećavanje datotečnog sustava (*ext3*, *ext4*) koji je formiran *nad* logičkim volume-om.
- Smanjivanje LVM-a (*shrink*) ide u tri faze:
  1. Provjera konzistentnosti i demontiranje (*umount*) datotečnog sustava;
  2. Smanjivanje datotečnog sustava (*ext3*, *ext4*) koji je formiran *nad* logičkim volume-om;
  3. Smanjivanje logičkog volumea, maksimalno za veličinu preostalog slobodnog prostora unutar grupe kapaciteta.



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# ZFS

- napredni datotečni sustav koji već dulji niz godina postoji na Solaris operacijskim sustavima
- ideja - objedinjavanje funkcionalnosti LVM-a, datotečnog sustava, upravljanja diskovima, korištenja *snapshots*, provjere integriteta (256-bit zaštitne sume), automatskog popravljavanja pogrešaka, korištenje naprednih RAID tehnologija (RAIDZ, RAIDZ2), ogroman teoretski kapacitet (ZB i veći), itd.
- implementacija ZFS-a pod Linuxom nije završena, ali se približava produkcijskoj fazi



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# XFS, I

- 64-bitni datotečni sustav koji je originalno razvijen od strane SGI-ja (*Silicon Graphics Inc.*)
- maksimalna duljina imena datoteke je 255 znakova, a u svojoj 64-bitnoj varijanti, podržava veličine datoteka do 16EB i veličine particija do 18EB
- kao kod *ext3* i *ext4*, radi se o *journaling* datotečnom sustavu.



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# XFS, II

- moguće je rezervirati brzinu za obavljanje operacija po XFS particijama (*Guaranteed Rate I/O*)
- prednosti pred ZFS-om – datotečni sustav koji postoji u produkcijskoj verziji na Linuxu, dok je ZFS još uvijek u testiranju.
- XFS je odnedavno postao i uobičajeni (*default*) datotečni sustav nove verzije RedHat-a, CentOS-a i Scientific Linuxa za sve particije osim */boot* particije (mjesto gdje se nalazi kernel i popratni sadržaji potrebni za uspješno pokretanje poslužitelja)
- većina korisnika smatra da je ovo bolji datotečni sustav od ext3/4 pri velikim opterećenjima, ali i uz veće zauzeće sistemskih resursa (naročito CPU)



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# Block-level i share-level pristup sustavu za pohranu

- kod *share-level* protokola, vidimo prostor na koji snimamo i sa kojeg čitamo podatke, ne vidimo datotečni sustav koji je na njemu niti napredne funkcije tog datotečnog sustava;
- kod *block-level* protokola (što direktno slijedi iz točke 1.), vidimo cijeli *block device* (LUN, disk), kojeg onda možemo formatirati datotečnim sustavom koji na tom uređaju želimo koristiti



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

# iSCSI

- iSCSI je jedan od protokola koji nudi *block-level* pristup mrežnim sustavima za pohranu
- koristi postojeću IP mrežu i sve njene tehnološke pretpostavke (IP adrese, portove, ...) za direktno povezivanje na LUN (*Logical Unit Number*, tj. block-level uređaj koji predstavljamo kroz iSCSI target portal)
- adresiranje u iSCSI-ju – koriste se IP adrese i IQN (*iSCSI Qualified Name*).
- IQN je zapravo iSCSIjev pandan mrežnim imenima (FQDN) iz klasičnih IP mreža
- klijentska strana (*initiator*) i poslužiteljska strana (*target*) imaju unikatni IQN, kao što imaju i unikatne IP adrese

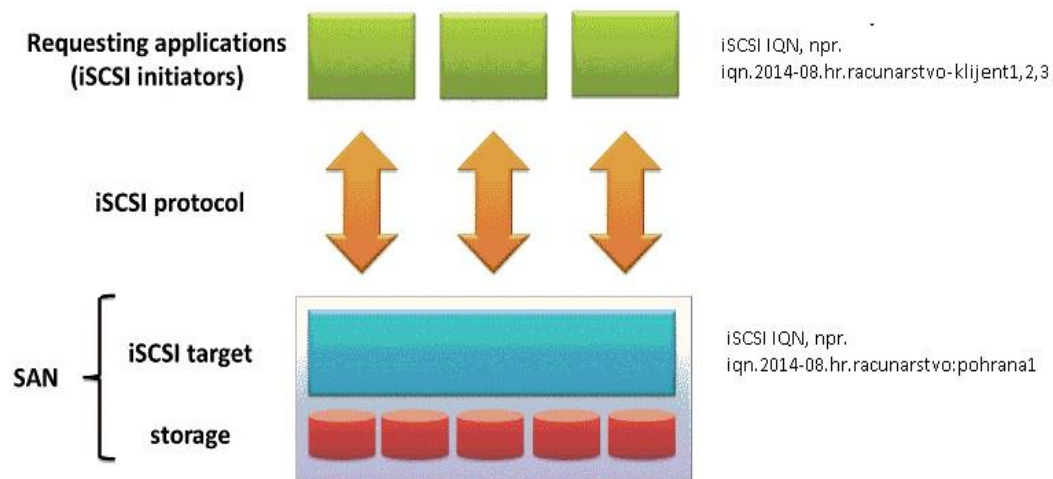


**Algebra**

visoka škola  
za primijenjeno  
računarstvo



# Komunikacija u iSCSI



**Algebra**

visoka škola  
za primijenjeno  
računarstvo



# iSCSI *target* i *initiator*

- target – „serverska” komponenta u iSCSI sustavu
- target portal – servis koji se „nalazi” na iSCSI uređaju na portu 3260 i nudi listu dostupnih LUN-ova iSCSI *initiatorima*
- initiator – „klijentska” komponenta u iSCSI sustavu



**Algebra**

visoka škola  
za primijenjeno  
računarstvo

?



**Algebra**  
visoko učilište