

Linux 2 DEVOPS 2020

Lektion 2

Uppvärmning

- Kommentarer / önskemål efter första dagen?
- Har alla funnit grupper att arbeta med för gruppuppgiften?

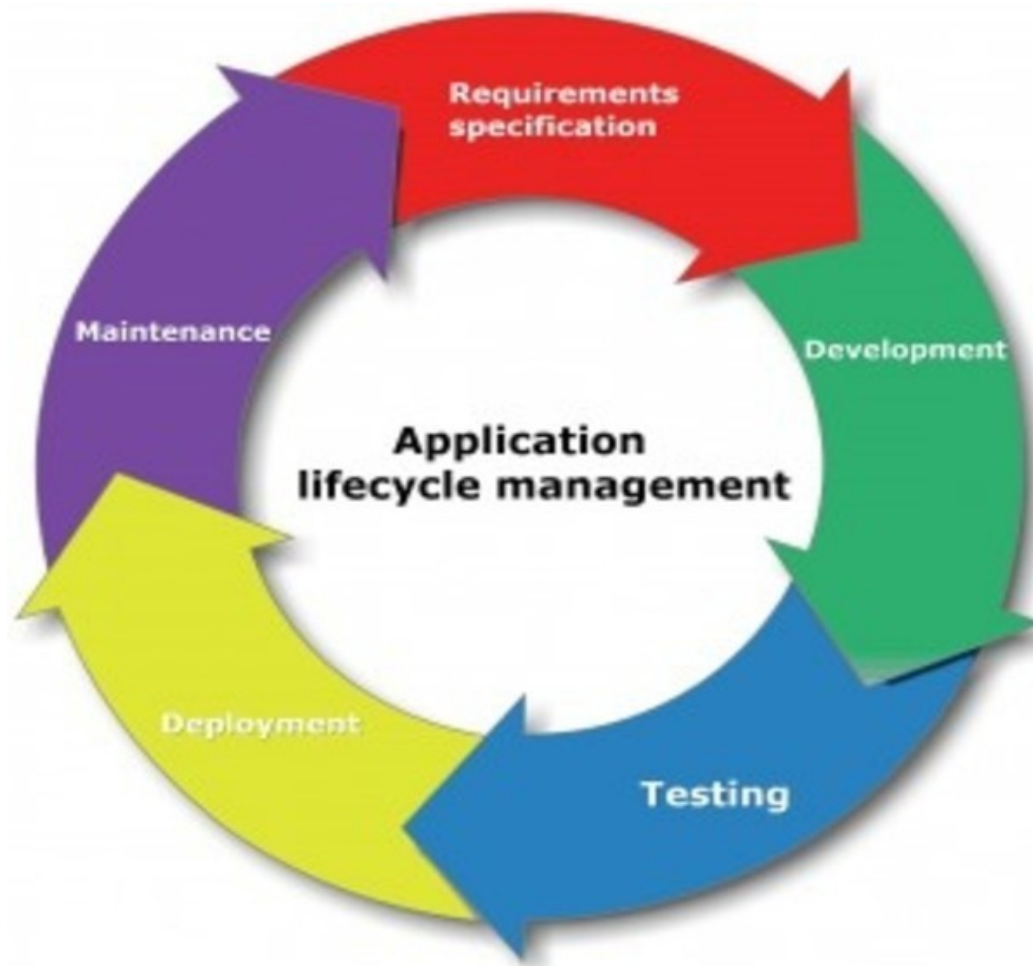
Idag

- Allmänt om server, uppbyggnad
- Miljöer för utveckling, test och drift
- Repetition: Filer och filrättigheter
- filöverföringar – om ftp, scp etc
- NFS, olika typer av filsystem

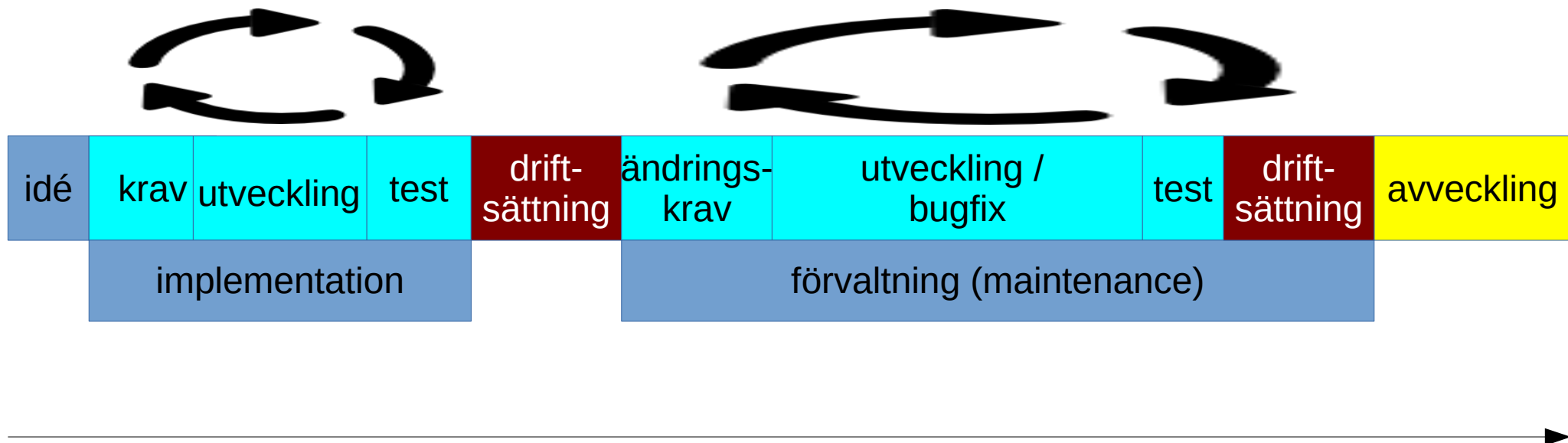
Linux-server

- Behöver först och främst veta vad just denna server är till för
- Använder sällan annat än kommandorad
- Fysiska och virtuella servrar

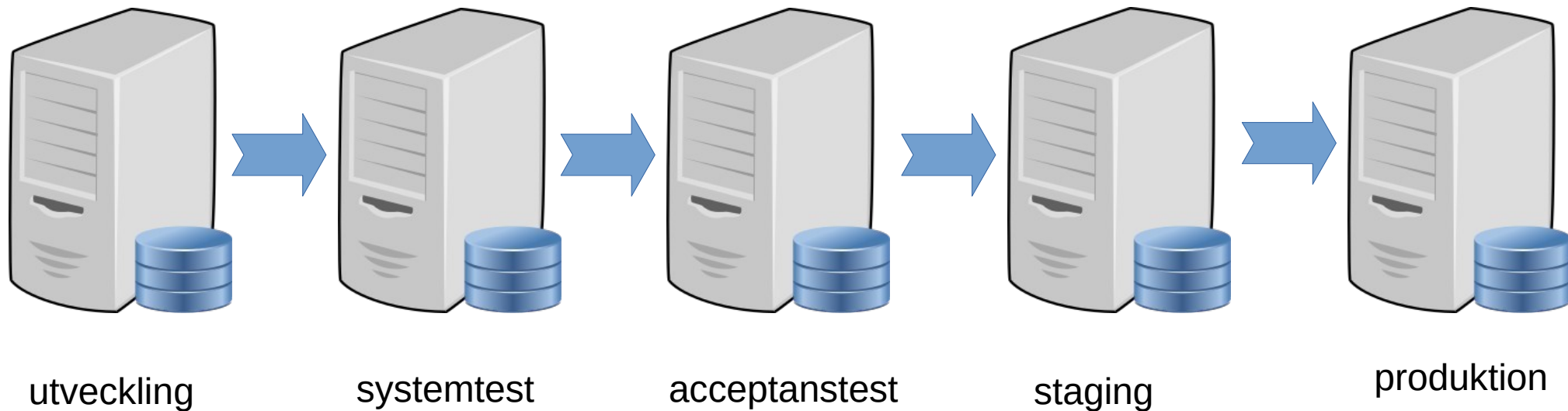
Applikationens livscykel



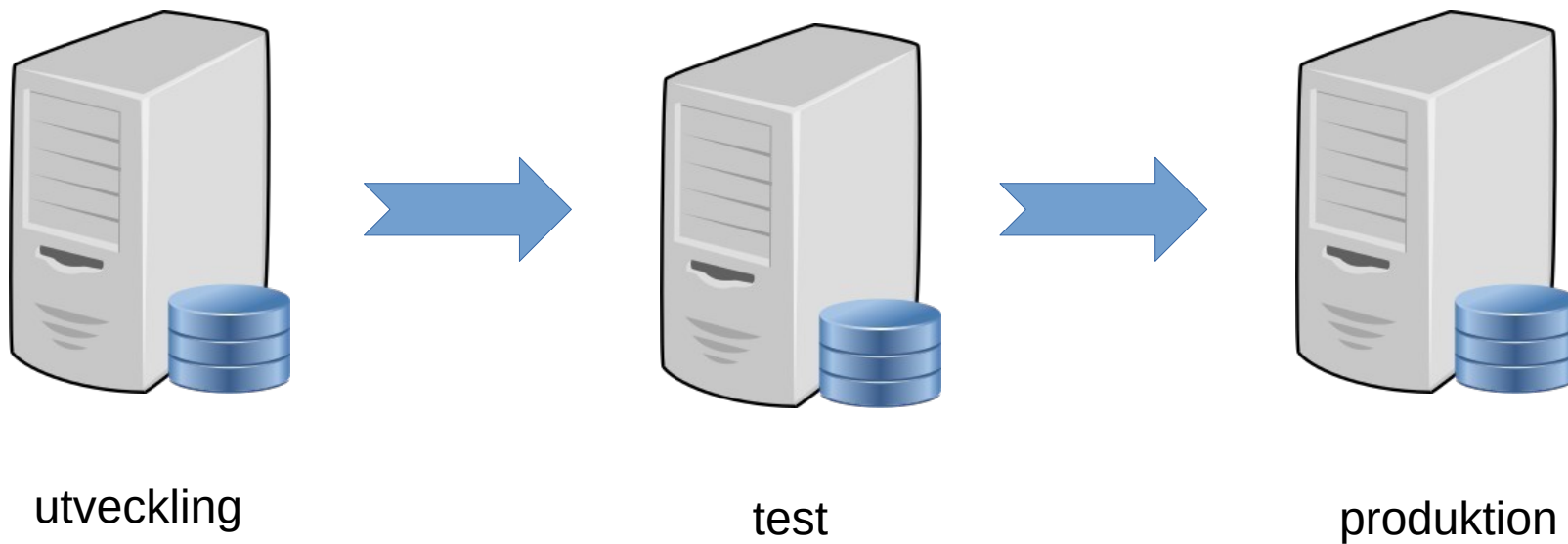
Tidslinje för ett system



Traditionell uppsättning miljöer



Mera minimalistisk uppsättning



Allmänt

- Ha inte saker installerade som inte behövs på Servrar – ha mindsetet “less is more”
- Koll på användare / vem som har tillgång till miljöerna
- Resursplanering utifrån serverns funktion

Utvecklingsmiljö

- Utvecklingsverktyg
- Koppling till versionshanterare
- Testdata
- Ofta på internt nätverk

Testmiljö

- Testverktyg
- Ofta även utvecklingsverktyg
- Testdata
- Ofta på internt nätverk

Driftsmiljö

- Enbart det som behövs för drift, dvs inga utvecklingsverktyg
- Äkta data
- Ordentlig, regelbunden backup
- Flera servrar för ökad driftssäkerhet - redundans
- Stark kontroll för tillgång endast för administratörer

Att ha koll på...

- Vilka program / paket behövs för det som skall finnas på servern?
- Vilka portar behöver vara öppna?
- Var i nätverket skall servern vara?
- Vilka skall ha tillgång till den?
- Vad för övervakning behövs?
- Vilka backuper behövs?

Övning 1

- Antag att ditt team utvecklar en Java-applikation med webbinterface, användardata i en MySQL-databas, samt ett antal informationsfiler som kan laddas ned från webben.
- Vad behöver ni på en utvecklingsserver respektive på en driftsserver? Fundera på alla verktyg som kan behövas.

Filer

- En Unix-princip är att allt är filer
- Filer har en ägare och rättigheter sätts på nivåerna ägare, grupp, övriga
- Filen har förstås en typ / ett format – till skillnad från i en del andra miljöer är det inte hårt kopplat till någonting i filens namn

`file <filename>`

- Verktyg för att hitta och hantera filer

Filrättigheter

- Rättigheter i tre set

-|user|group|world|

-|rwx|rwx|rwx|

Exempel: `-rwxrwxr-x 1 niklas niklas 0 aug 9 19:53 fil`

- Tripplerna kan också uttryckas i siffror

-|421|421|421|

adderas för att ge rättigheter per kategori, exemplet med allt "på" blir 777, exemplet med filen ovan blir 775

Ändra filrättigheter

- **chown** – byt ägare
`chown foo fil1`
- **chgrp** – byt grupp
`chgrp bar fil1`
 - Eller ändra ägare och grupp samtidigt
`chown foo:bar fil1`
- **chmod** – ändra rättigheter
`chmod u+rwx fil1`
`chmod g-wx fil1`
`chmod o+x fil1`
`chmod 764 fil1`

Filrättigheter

- ...och den där första positionen
 - "-" för en vanlig fil
 - "d" för ett directory
 - "c" för "character special", en device
- "Sticky bit"
 - Hindrar andra än ägaren från att manipulera filen

Exempel: `drwxrwxr-t 2 niklas niklas 4096 aug 9 19:58 foobar`

 - `chmod +t <fil>`

Kopiera, flytta osv

- cp – kopiera filer
cp fil1 fil2
cp -r dir1 dir2
- mv – flytta filer
mv fil1 fil2
- rm – ta bort filer
rm fil1
rm -r dir1
- touch – markera en fil som läst
touch fil1

Övning 2

- Gör ett directory filexempel, lägg in filer med namnen test1 ... test10 i det (använd gärna en loop).
- Gör ett script som hittar alla filer i det directory där det körs som har namn som börjar med test. Låt det kopiera varje sådan fil till en som har samma namn med tillägget ".bak".
- Testa att köra scriptet som olika användare. Testa sedan att sätta sticky bit på directory filexempel och se hur det blir när ni försöker köra scriptet som olika användare.

Övning 2

```
for i in {1..10}  
do  
  touch test$i  
done
```

Övning 2

```
#!/bin/bash
```

```
for i in `ls test*`  
do  
  cp $i $i.bak  
done
```

Skriv till filer

- Skriv över fil1:
`echo "hej" > fil1`
- Lägg till i slutet av fil1 (append):
`echo "hej" >>fil1`
- Skicka stderr till en fil med 2>
`echo "hej" >>fil1 2>&1`
- Input från fil1:
`cat <fil1`

Kom ihåg regexp

- Användbart för att hitta strängar och för att byta ut strängar
- Exempel (från Linux 1):
`${arr[*]//rad//foo}`
- Ytterligare verktyg: sed
 - Exempel `cat fil1 | sed 's/abc/ABC/g'`

Övning 3

- Skapa en fil fil1 med ett antal namn:

Tintin

Milou

Haddock

Kalkyl

Castafiore

Nestor

Dupond

Dupont

- Gör ett skript som går igenom filen, byter ut "Milou" mot "Milou Hund" och skriver resultatet till en fil fil2.
- Extraövning: Gör nu samma sak på (minst) ett annat sätt.

Övning 3

```
#!/bin/bash
```

```
while read a; do  
    echo ${a//Milou/Milou Hund}  
done < fil1 > fil2
```

Variant med sed

- Som scriptet nyss:

```
sed 's/Milou/Milou Hund/g' <fil1 >fil2
```

- För att enkelt göra utbytet i samma fil:

```
sed -i -e 's/Milou/Milou Hund/g' fil1
```

Hitta filer

- Find

- Exempel

- `find . -name "*sh"`

- `find . -newermt "2020-08-10"`

Övning 4

- Gör ett script som hittar alla filer som heter något som slutar på "sh" (i det directory där scriptet körs plus underdirectories) och som skriver ut resultatet av ett `ls -l` för de filerna.

Övning 4

```
#!/bin/bash
```

```
for i in `find . -name "*sh"`  
do  
  ls -l $i;  
done
```

Filöverföringar

- ftp – osäkert, skickar allt i klartext
- scp – tänk cp över ssh

```
scp file remote_user@remote_ip:/remote/directory
```

```
scp remote_user@remote_ip:/remote/directory/file file
```

Filöverföringar

- sftp – tänk säkrad ftp

```
sftp remote_user@remote_ip
```

- Några sftp-kommandon

```
ls
```

```
cd <directory>
```

```
get <file>
```

```
put <file>
```

```
bye
```


Övning 5

- Testa att föra över en fil med scp respektive med sftp. Det går att köra mot localhost.
 - Vad behöver du ha igång på din dator för att det skall fungera?

Andra filöverföringar

- wget – enkel hämtning från en webbserver med http / https (eller ftp)

```
wget https://www.dn.se/
```

```
wget -O test.html https://www.dn.se/
```

- curl – verktyg som kan hantera många olika protokoll

```
curl https://www.dn.se/
```

```
curl -X POST http://www.yourwebsite.com/login/ -d  
'username=yourusername&password=yourpassword'
```

Övning 6

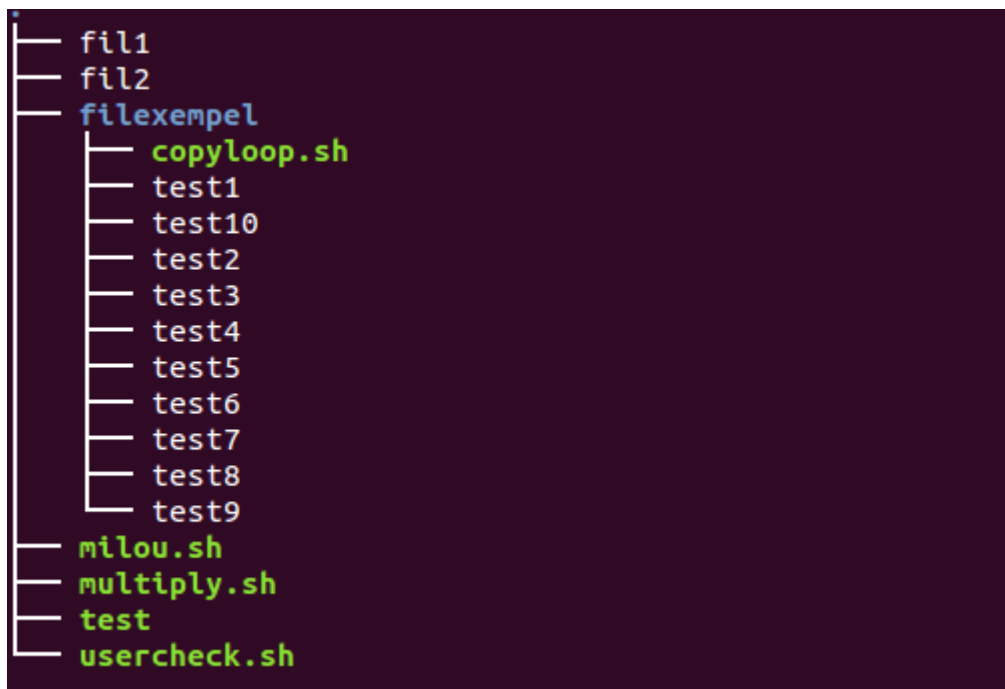
- Hämta data från `www.nackademin.se` med `wget`.
- Hämta data från `www.nackademin.se` med `curl`.
- Kan ni logga in i studentportalen med `curl`?

Från filer till filträd...

- tree – ett verktyg för att enkelt visualisera filträd

apt install tree

tree -a .



```
tree -a .
.
├── fil1
├── fil2
├── filexempel
│   ├── copyloop.sh
│   ├── test1
│   ├── test10
│   ├── test2
│   ├── test3
│   ├── test4
│   ├── test5
│   ├── test6
│   ├── test7
│   ├── test8
│   └── test9
├── milou.sh
├── multiply.sh
├── test
└── usercheck.sh
```

tree

- Begränsa hur många nivåer ned den söker sig:
 - L <antal>
- Enbart directories:
 - d
- Ta med hela sökvägen:
 - f
- Få med filrättigheterna:
 - p

Övning 7

- Installera tree (om du inte redan har den).
- Titta på vad som ligger på din Linux-burk genom att göra ett "träd" från /
 - Tips: Det blir betydligt enklare och tar mindre lång tid om du nöjer dig med att titta på första nivån
- Gör nu samma sak som nyss men med alla filrättigheter utskrivna

Övning 7

```
tree -L 1 /
```

```
tree -L 1 -p /
```

... till typer av filsystem

- Det finns olika sätt att lagra data. I Linux finns det stöd för runt 100 olika filsystemstyper.

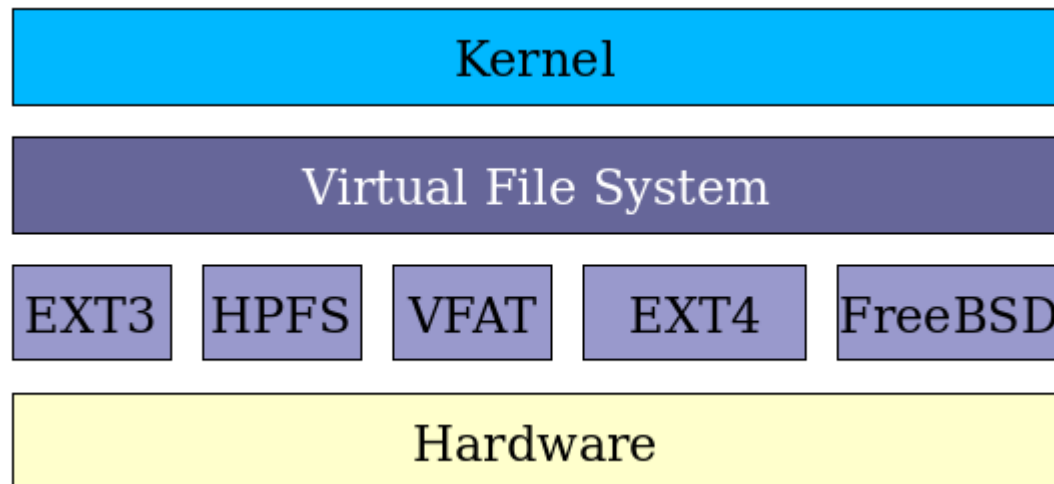


Bild lånad från opensource.com

Filsystem

- mount för att göra ett filsystem tillgängligt
 - En "mount point" för var i filträdet det kommer finnas
 - Till exempel en disk särskilt för /opt/data
- /etc/fstab

Typer av filsystem

- Journaling file system

"A journaling file system is a file system that keeps track of changes not yet committed to the file system's main part by recording the intentions of such changes in a data structure known as a "journal", which is usually a circular log. "

(Wikipedia)

- Exempel: Ext4, ZFS, FAT, ReiserFS

Typer av filsystem

- Versioning file system

”A versioning file system is any computer file system which allows a computer file to exist in several versions at the same time. Thus it is a form of revision control.” (Wikipedia)

- Exempel: NILFS

Typer av filsystem

- Distributed file system

”Distributed file systems do not share block level access to the same storage but use a network protocol. These are commonly known as network file systems, even though they are not the only file systems that use the network to send data”
(Wikipedia)

- Exempel: NFS

Typer av filsystem

- Default filsystemstyp i de flesta moderna Linux-system: ext4
- Protokoll för mount av externa filsystem: nfs

```
mount -t nfs <ip_address>:<filedir> <dir>
```

```
mount -t nfs 10.10.0.10:/backups /var/backups
```

Tillbakablick, reflektion, kommentarer ...