

# Linux 2 DEVOPS 2020

## Lektion 1

# Presentation

Niklas Engvall

Niklas.engvall@iver.se

Marcus Jonsson

Marcus.jonsson@iver.se

William Kvamme

William.kvamme@iver.se

# Om kursen

- Fortsättning på Linux 1
- Shell scripts, filer, installationer, paket, nätverk, virtuella miljöer
- Server-fokus, kommandorad
- Alla förväntas ha en egen Linux-installation att labba med
  - Exempel mestadels på Ubuntu

# Om kursen

- Undervisning på distans – Zoom
  - Måndag, tisdag, torsdag
- Examination i två delar:
  - Gruppuppgift
  - Skriftlig tentamen

# Om kursen: typisk kursdag

- Förmiddag: Genomgångar med pauser och korta övningar
  - Övningar tillsammans i "break out rooms"
  - Korta övningar var och en för sig
- Eftermiddag: Någon genomgång med övning (oftast), repetition, tid att jobba ihop
- Veckans sista kursdag virtuell AW sista timmen

# Idag

- Repetition allmänt om Linux
- Repetition om användare
- Repetition / bli varm igen grunderna för shell scripts
- Genomgång av gruppuppgiften

# Unix family Tree

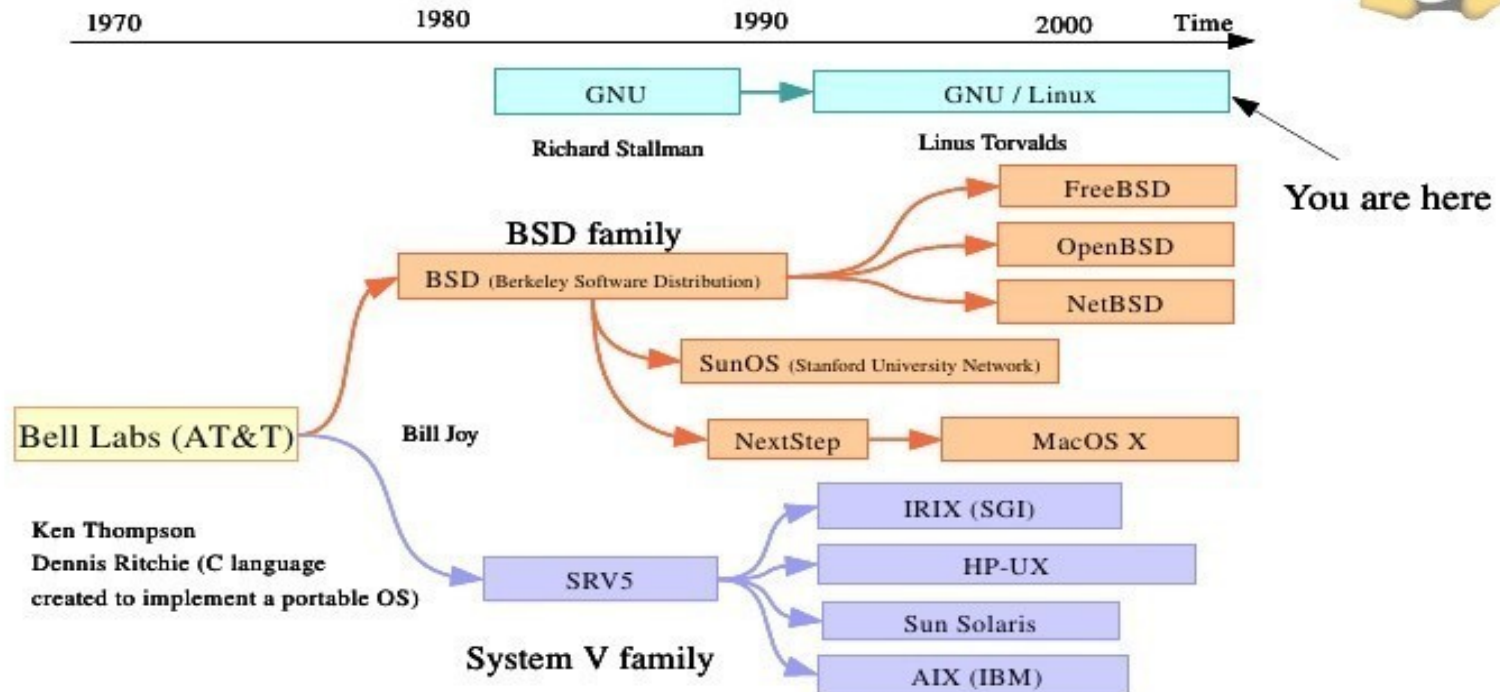
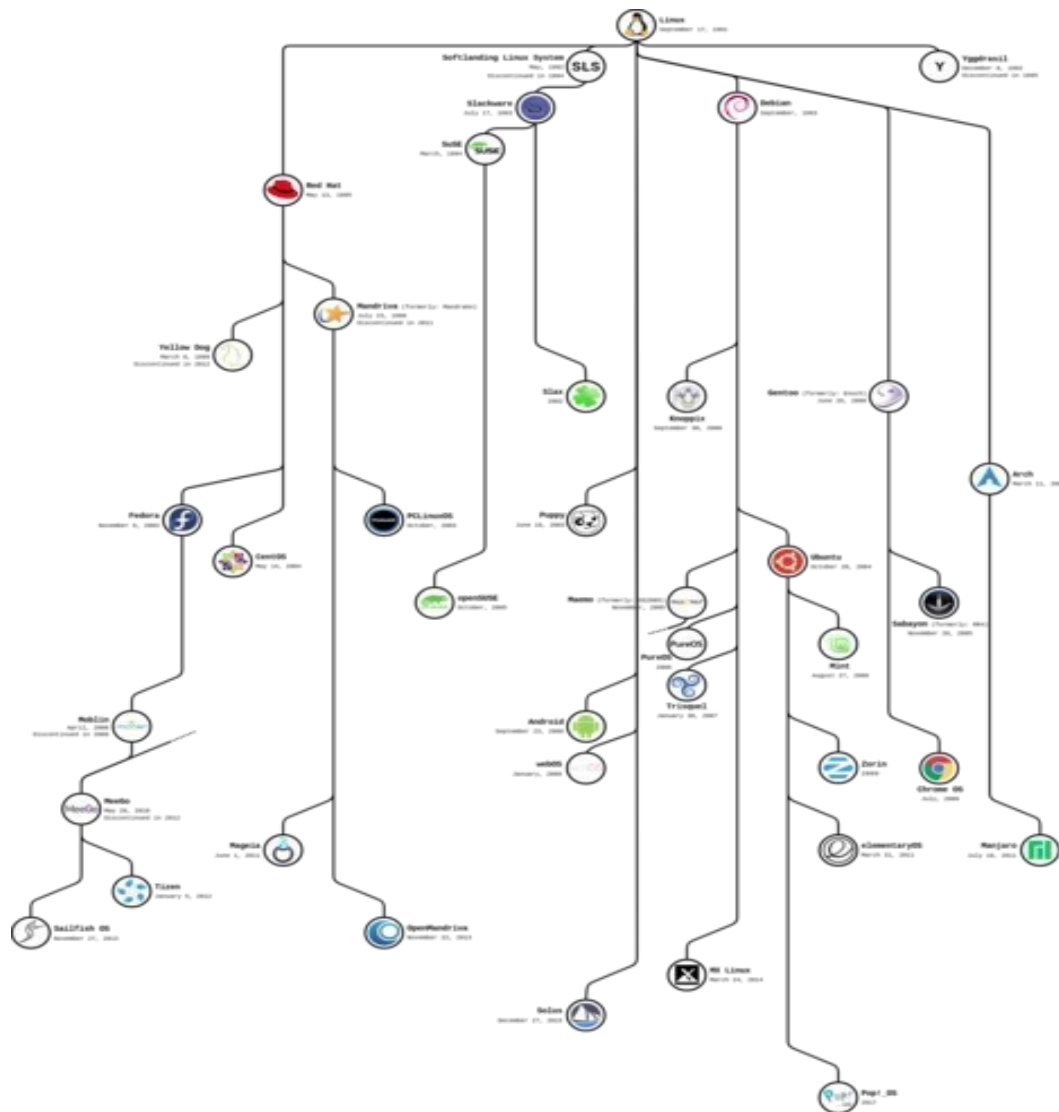


Bild knytt från FOSS History



Linux distros, bild från Wikipedia



# Unix-/Linux-principer

- Specialiserade program för att göra saker, sätts ihop för att åstadkomma större saker

`ls -l | grep test`

- Allt är filer
- Allt kan anpassas

# Användare

- Loggar in
- Användarnamn, lösenord
- whoami
- /etc/passwd
- /etc/shadow

# Övning 1

Uppvärmningsövning:

- Logga på din Linux-installation
- Se ditt användarnamn med whoami
- Hitta dig själv i /etc/passwd (kommer ni ihåg grep?)
- Hitta dig själv i /etc/shadow (här behöver du se till att ha rätt att läsa i den)

# Övning 1

```
> grep mo /etc/passwd
```

```
niklas:x:1000:1000:Niklas Engvall:/home/niklas:/bin/bash
```

```
> sudo grep niklas /etc/shadow
```

```
niklas:
```

```
$6$GKKkA67j$EgFKNA904ycLM.LwVIMcFEjmYpef4ohkw.  
nCHecAGrnk4EeoD05VY690NqgL95LjR0Tz5wyq4NddQvi  
H1mOnl1:16190:0:99999:7:::
```

# /etc/passwd

oracle:x:1021:1020:Oracle user:/data/network/oracle:/bin/bash

The diagram shows the entry 'oracle:x:1021:1020:Oracle user:/data/network/oracle:/bin/bash' with arrows pointing from specific fields to numbers 1 through 7. The fields are: 'oracle' (1), 'x' (2), '1021' (3), '1020' (4), 'Oracle user' (5), '/data/network/oracle' (6), and '/bin/bash' (7).

Field	Number
oracle	1
x	2
1021	3
1020	4
Oracle user	5
/data/network/oracle	6
/bin/bash	7

1. Username
2. Password:
3. User ID (UID)
4. Group ID (GID)
5. User ID Info
6. Home directory
7. Command/shell

Bild från  
[www.cyberciti.biz](http://www.cyberciti.biz)

# Lägg till användare

- adduser

`sudo adduser <username>`

- Ange ett lösenord

`sudo useradd <username>`

`sudo passwd <username>`

# Användargrupper

- /etc/group
- En användare har en primär grupp och inga eller flera sekundära grupper

```
sudo useradd -a -G <groupname> <username>
```

```
sudo useradd -g <groupname> <username>
```

```
sudo groupadd <groupname>
```

# Övning 2

- Titta i `/etc/group`
- Hitta alla grupper som din användare tillhör



# Övning 3

- Nu gör vi ett litet script...
- Gör ett script `usercheck.sh` som tar en inparameter och letar efter denna som användarnamn i `/etc/passwd` och `/etc/group`, samt skriver ut de raderna
  - Användning exempel: `> usercheck.sh niklas`
  - Det får vara grovt och hitta även andra rader än de man är ute efter

# Övning 3

```
#!/bin/bash  
# simple example script  
# usage: usercheck.sh <username>
```

```
user=$1;  
echo "from passwd:";  
grep $user /etc/passwd;  
echo "from group:";  
grep $user /etc/group;
```

# Superhjältar... ehum, -användare

- Root
- Sudo
- Testa...  
    sudo whoami
- Sudoers och sudo group  
    /etc/sudoers

# Superanvändare

- Ge sudo-rättigheter

`usermod -a -G sudo <username>`

`usermod -aG sudo <username>`

- Alternativt (ej rekommenderat, använd `/etc/sudoers.d/` och skapa filer som ger rättigheter):

`visudo`

(för att editera `/etc/sudoers`)

# Köra som annan användare

- Går förstås att göra även om man inte skall bli root

- su

su <username>

su - <username>

# Övning 4

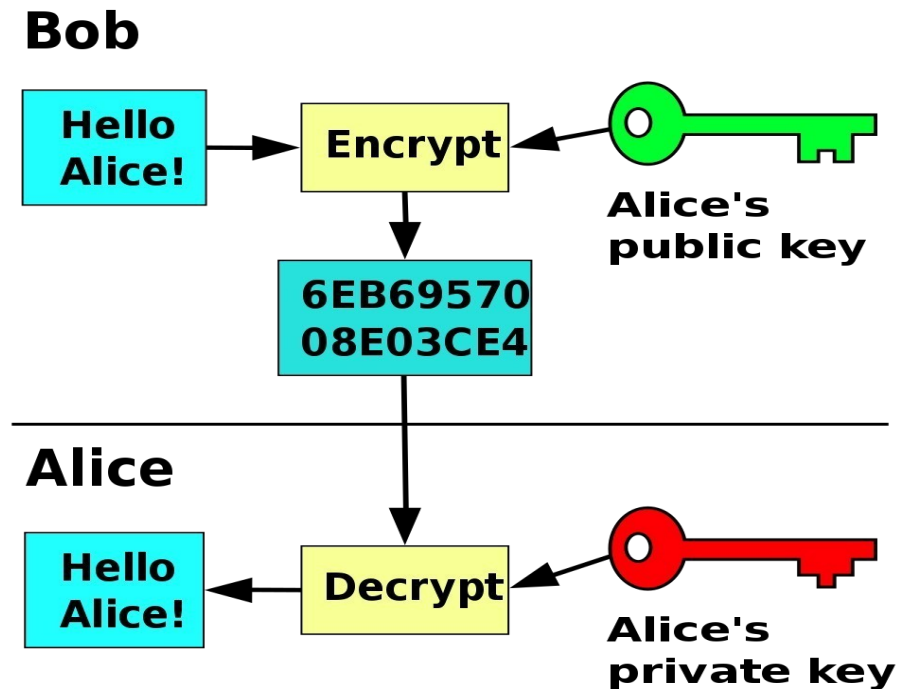
- Skapa en ny användare
- Observera vad den användaren får för grupptillhörighet per default
- Lägg till användaren i gruppen sudo
- Testa så du kan göra “su” till den användaren och sedan köra något som sudo.  
(t ex sudo whoami)

# ssh

- ssh (secure shell) – protokoll för att kommunicera med dator, oftast över nätverk
  - Om du inte har sshd igång, installera och starta den!  
`sudo apt install openssh-server`  
`systemctl start sshd`
  - Testa: `ssh localhost`
- "Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. Typical applications include remote command-line, login, and remote command execution, but any network service can be secured with SSH." (en.wikipedia.org)

# ssh

- Använder sig av publik nyckel-kryptering
- PKI – Public Key Infrastructure





# ssh

- Använder sig av publik nyckel-kryptering
- ssh-keygen för att skapa nyckelpar (rsa är standard)
  - Utöver RSA finns: DSA, ECDSA, Ed25519
- Defaultport 22
- man ssh för att hitta diverse varianter
  - Exempel: ssh <server> -p <port> -l <login>
- Login med lösenord eller som förlitar sig på att din privata nyckel matchar den publika nyckeln på servern

# Script-repetition

- bash
- Ge värde till en variabel: `var=10`
- Använd variabelvärdet: `$var`
- Exempel:  
    `var=10`  
    `echo $var`

# Script-repetition

- Aritmetik: (( ))
  - Variabler behandlas som strängar, men man kan räkna genom användning av dubbelparanteser
  - Exempel: `sum=$((var*2))`
- for-loop: `for <var> in <list>`
  - Kan också vara en räknare med heltal
  - Exempel: `for name in `ls``
  - Exempel: `for i in {1..5}`
- `for – do – done`

# Övning 5

- Gör ett skript som skriver ut multiplikationstabellen 1 till 10
- Använd for-loopar

# Övning 5

```
#!/bin/bash

#simple multiplication tables

for i in {1..10}
do
  for j in {1..10}
  do
    res=$((i*j));
    echo $i "*" $j "=" $res;
  done
done
```

# Script-repetition

- Villkor:
  - = or == is equal to
  - != is not equal to
  - < is less than in ASCII alphabetical order
  - > is greater than in ASCII alphabetical order
  - z test that the string is empty (null)
  - n test that a string is not null
- Exempel: [ \$a > \$b ]

# Script-repetition

- if – then – else
- if [villkor] then /.../ fi
- Notera att villkor kan vara av väldigt olika slag

# Övning 6

- Lägg till i scriptet från övning 5 att när de två tal som multipliceras är lika, skriver det "hej"



# Övning 6

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

```
#simple multiplication tables
```

```
for i in {1..10}
do
  for j in {1..10}
  do
    res=$((i*j));
    echo $i "*" $j "=" $res;
    if [ $i == $j ]
    then
      echo "hej";
    fi
  done
done
```

Tillbakablick, reflektion, kommentarer ...  
... sedan skall vi prata om gruppuppgiften.

# Gruppuppgiften

- Uppgiften utförs i grupp, helst 3-4 personer
- Att genomföra uppgiften är ett krav för godkänd kurs
- De som gör en godkänd redovisning på någon av de avsatta tiderna sista kursveckan får dessutom 2 bonuspoäng på tentan
  - Sikta på att redovisa måndag 15/2!
  - Reservtider för redovisning tisdag 16/2 (förmiddag)

# Gruppuppgift

- Scenario: Ni skall bygga upp miljöer för både utveckling / test och drift av en enkel applikation
  - Operativsystemet är Linux
  - Applikationskoden kommer hanteras i git
  - Miljöerna kräver apache webserver och MySQL
  - Ni får själva uppfinna / anta vad som kan behövas för övrigt
- Ni skall göra ett förslag till hur miljöerna skall byggas och hur driftsättning från test skall gå till

# Gruppuppgift

- Ni skall visa hur ni hanterar användare i respektive miljö
- Driftsättning bör automatiseras i rimlig mån
- Ni skall lägga upp någon form av övervakning av väsentliga program i drift
- Det finns många sätt att lösa den här uppgiften – alla fungerande sätt blir godkända

# Grupppuppgift

- Redovisningen skall presentera resonemang om hur ni valt era miljöer och varför
- Redovisningen skall innehålla ett element av riskanalys
- Redovisningen skall visa en demo av hur ni byggt miljöerna
- Redovisningen bör ta 10-15 minuter per grupp