

# Linux 2

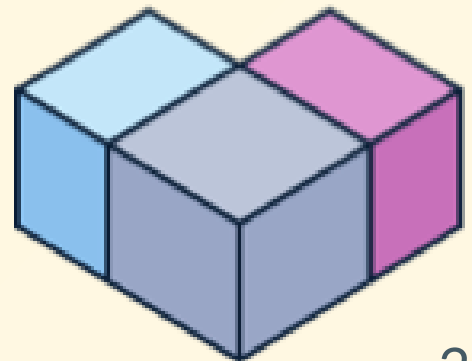
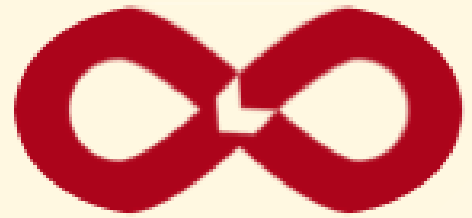
# DEVOPS 2022

## Lektion 1

# Presentation: Nevyn Bengtsson

## Bakgrund

- Spelprogrammering, BTH (3-årig utbildning)
- Spotify AB: iPhone-apputveckling
- Lookback Inc.: Grundare, CTO
- Alloverse AB: Grundare, CEO, CTO



# Presentation: Nevyn Bengtsson

## Utbildningserfarenhet

- CoderDojo, MobileBridge, hobbyistlärare
- Mitt första lärarjobb!

# 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 är mestadels på Ubuntu

# Om kursen

## Undervisning på plats

Måndag, onsdag, torsdag

## Examination i två delar

- Gruppuppgift
- Skriftlig tentamen

# Om kursen

## Förmiddag

Genomgångar med pauser och korta övningar. Övningarna både i grupp och var för sig.

## Eftermiddag

Någon genomgång med övning (oftast), repetition, tid att jobba ihop

# Om kursen

## Tider

0900–1530

(Om jag kan gå en halvtimme tidigt så kan jag svara på frågor en stund på kvällen)

# Dagens kurs



# Dagens kurs

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

1970

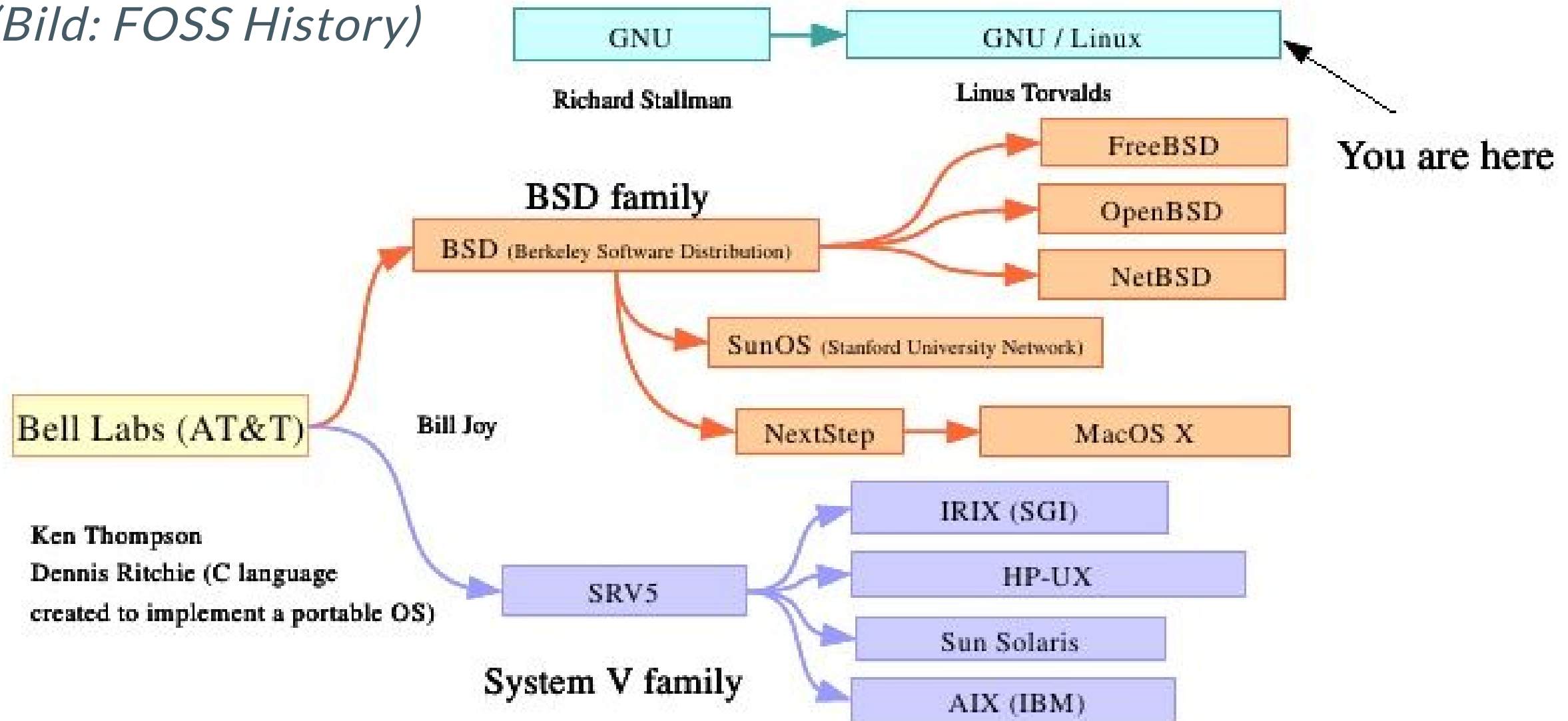
1980

1990

2000

Time

(Bild: FOSS History)





Debian



debian

*(Bild: Cognitive Waves)*

Ubuntu



ubuntu

SLS

Slackware



slackware  
linux

S.u.S.E

SuSE



SUSE

openSUSE

SLES



CentOS

CentOS

Red Hat

Red Hat Enterprise



redhat

Fedora Core

Fedora



# Unix-/Linux-principer

- "Do one thing well": Specialiserade program för att göra små, väldefinierade saker
- "Write programs to work together": sätt ihop enkla program för att åstadkomma större saker:
  - `ls -l | grep test`
- Allt är filer
- Allt kan anpassas

# Användare

# Användare

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

# Övning 1

Uppvärmningsövning:

1. Logga på din Linux-installation
2. Se ditt användarnamn med whoami
3. Hitta dig själv i `/etc/passwd` (kommer ni ihåg grep?)
4. 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
H1m0nI1:16190:0:99999:7:::
```



# /etc/passwd

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

1 2 3 4 5 6 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

- `sudo adduser <username>`
- `sudo useradd <username>` --  skapar inte hem-mapp, etc!
- `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 editera /etc/sudoers direkt (*ej rekommenderat, använd /etc/sudoers.d/ och skapa filer som ger rättigheter*):

```
visudo
```



# Köra som annan användare

(Framförallt för att bli root, men funkar även med andra användare)

```
su <username>      # behåll environment  
su - <username>    # nollställ environment
```

# 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)

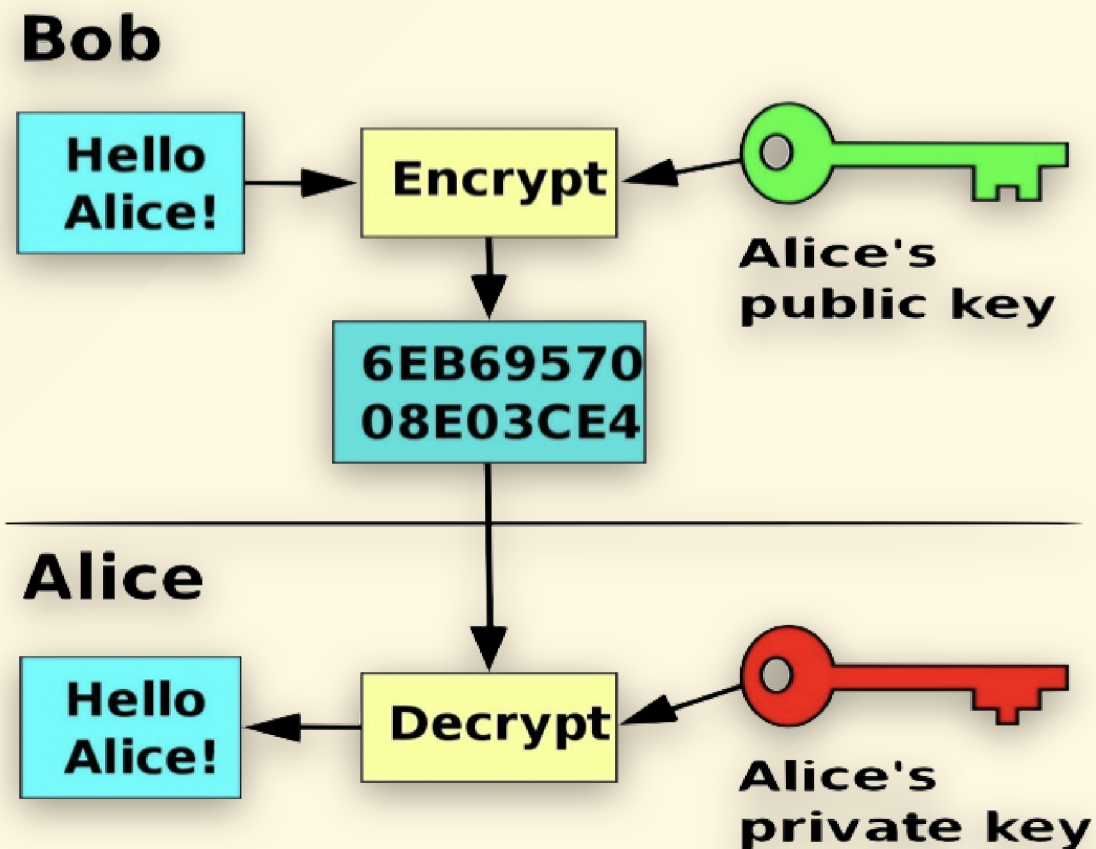
# ssh

# ssh

- 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

# 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

# ssh

- `man ssh` för att hitta diverse varianter
  - Exempel: `ssh <server> -p <port> -l <login>`
- Två olika sätt att logga in:
  - Lösenord (inte rekommenderat)
  - Login med privat nyckel som matchar den som finns på servern (rekommenderat)
- `~/.ssh/authorized_keys`

# Script-repetition



# 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.

# Gruppppppgift

# 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 tisdag 13/9!
  - Reservtider för redovisning onsdag 14/9 (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

# Gruppuppgift

- 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