

Linuxadministration I 1DV417 - Laboration 6  
NIS och bash-scriptning

Marcus Wilhelmsson  
marcus.wilhelmsson@lnu.se

22 februari 2013

## Innehåll

<b>1</b>	<b>Inledning och mål</b>	<b>3</b>
<b>2</b>	<b>Material och genomförande</b>	<b>3</b>
<b>3</b>	<b>Förberedelseuppgifter</b>	<b>3</b>
<b>4</b>	<b>NIS</b>	<b>3</b>
4.1	Kriterier . . . . .	3
4.2	Kontroll av funktionalitet . . . . .	4
<b>5</b>	<b>bash-programmering</b>	<b>4</b>
5.1	Ett enkelt script . . . . .	4
5.2	Scripta in användare . . . . .	5
5.3	Scripta en filstruktur . . . . .	5
5.4	In- och utmatning . . . . .	6
<b>6</b>	<b>Laborationsfrågor</b>	<b>6</b>

# 1 Inledning och mål

För att underlätta administrationen av ett Linux/UNIX-baserat nätverk kan tjänsten NIS användas. NIS tillhandahåller verifiering av användare från en eller flera centrala servrar vilket underlättar administreringen av användarna.

Ett flertal av de dagliga uppgifterna en administratör ställs inför kan automatiseras med hjälp av script. Du kommer därför i denna laboration att lösa ett antal problem genom script-programmering i bash.

Efter genomgången laboration kommer du ha praktisk kännedom om hur du:

- konfigureras en NIS-domän.
- förenklar några vanligt förekommande systemadministrationsuppgifter genom script.

# 2 Material och genomförande

Laborationen kommer genomföras i VMware Workstation som tillhandahåller möjligheten att köra flera virtuella datorer i en fysisk. Utför laborationens uppgifter och moment samt dokumentera vad du kommer fram till på de olika delarna. Vid redovisning av laborationen ska du med hjälp av laborationsrapporten på ett komplett sätt redovisa *vad* du har gjort, *hur* du har gjort det samt *vad* du kom fram till och *varför* du kom fram till det du gjorde.

# 3 Förberedelseuppgifter

Laborationen har en förberedelseuppgift. Se till att du genomfört förberedelseuppgiften innan du påbörjar laborationen då du med hjälp av förberedelseuppgiften ska genomföra laborationen.

1. är du osäker på syntaxen i bash? Du måste sätta dig in i denna innan laborationen.
2. Läs följande kapitel i boken:
  - Kapitel 2
  - Kapitel 19, sid. 736 – 738
  - Kapitel 22, sid. 935 – 937

# 4 NIS

Du ska i denna uppgift sätta upp en NIS-domän för att underlätta administreringen av användarkonton i ett nätverk genom att användarkontona sparas och administreras från en central knutpunkt.

## 4.1 Kriterier

Följande kriterier ska uppfyllas:

1. Två NIS-servrar ska finnas, en primär och en sekundär. Dessa ska finnas på din personliga del av CSLab-nätverket med lämpliga namn och nätverksinställningar. Lägg även till dem i ditt DNS-system.
2. NIS-domännamnet ska vara satt till ditt användarnamn.

3. Endast datorer från ditt IP-nät i CSLab-nätverket ska få komma åt serverna.
4. När maps-databasen uppdateras på den primära servern ska dessa ändringar automatiskt laddas ner av den sekundära servern.
5. En klient ska finnas och den ska använda NIS för inloggning. Se till att både den primära och den sekundära kan användas för att logga in, detta för att skapa redundans.
6. Användares hemkataloger ska exporteras via NFS från den primära NIS-servern. Glöm ej att sätta `USE_IDMAP=yes` i `/etc/default/nis`. Klienten ska montera de exporterade hemkatalogerna automatiskt.

## 4.2 Kontroll av funktionalitet

Genomför följande kontroller och tester.

1. Lägg till en ny användare på den primära NIS-servern.
2. Uppdatera NIS-databasen på den primära NIS-servern.
3. Kontrollera att användaren replikerats till den sekundära NIS-servern.
4. Logga in med användaren från klienten.

## 5 bash-programmering

Du ska nu undersöka hur script skrivna i bash kan underlätta administreringen av ett Linux-system.

### 5.1 Ett enkelt script

För att du ska komma igång med bash-scriptningen ska du i den första uppgiften skapa ett enkelt script som beroende på klockan skriver ut olika meddelanden när det körs. Du kommer endast att behöva en dator under denna uppgiften.

1. Skapa en scriptfil med följande funktionalitet:
  - Om klockan är mellan 05:00 – 10:59 ska meddelandet *morning* skrivas ut.
  - Om klockan är mellan 11:00 – 12:59 ska meddelandet *lunch* skrivas ut.
  - Om klockan är mellan 13:00 – 16:59 ska meddelandet *afternoon* skrivas ut.
  - Om klockan är mellan 17:00 – 20:59 ska meddelandet *evening* skrivas ut.
  - Om klockan är mellan 21:00 – 04:59 ska meddelandet *night* skrivas ut.
  - Tänk på att du inte behöver räkna minuter, bara timmar.
2. För att byta klockslag på datorn skriver du: `date [månad][dag][timme][minut]`  
Ex: `date 02131500` blir då den 13:e februari kl 15:00
3. Testa scriptet.

## 5.2 Scripta in användare

I detta moment ska du från ett bash-script skapa användare till systemet. Informationen om användarna ska läsas in från filen *users* som du ska skapa. Nedan beskrivs de poster som ska finnas i filen.

- användarnamn
  - förnamn
  - efternamn
  - standardgrupp (standardgruppen för alla användare ska vara *labusers*)
  - lösenord (tänk på att useradd -p lägger till lösenordet i klartext, titta på kommandot mkpasswd för att kryptera lösenordet)
1. Skapa *users*-filen.
  2. Skapa ett script som läser in *users*-filen och skapar användarkontona med hjälp av informationen i filen. Följande krav gäller för scriptet:
    - När en användare läggs till ska scriptet undersöka om allt gick bra och i sådana fall gå vidare.
    - Om det inte går att lägga till ett användarkonto ska scriptet skriva ut ett meddelande om vilken användare som inte kunde läggas till. Ett tips är att titta på returvärdet.

## 5.3 Scripta en filstruktur

I detta moment ska du skapa katalogstrukturen du ser nedan:

```
/
├── shared/
│   ├── users/
│   │   ├── kalle/
│   │   ├── olle/
│   │   └── nisse/
│   ├── applications/
│   └── projects/
│       ├── lnu/
│       ├── uos/
│       └── development/
│           └── homepage/
```

Strukturen skapas med hjälp av ett bash-script. Det är inte tillåtet att i scriptet statiskt skapa katalogstrukturen, man ska alltså kunna använda vilken katalogstruktur som helst tillsammans med scriptet. Katalogstrukturen ska läsas in från en textfil för att sedan skapas med hjälp av scriptet.

```
shared users kalle
shared users olle
shared users nisse
shared applications
shared projects
shared projects lnu
shared projects uos
shared projects development
shared projects development homepage
```

1. Skapa ovanstående textfil i valfri katalog.
2. Skapa ett bash-script som med hjälp av ovanstående fil skapar katalogstrukturen som beskrivs i uppgiften. Observera att scriptet ska kunna hantera större katalogstrukturer än den som är beskriven ovan. Om en katalog redan finns ska scriptet gå in i katalogen och därefter skapa underkatalogen.

## 5.4 In- och utmatning

Du ska i detta moment skapa ett bash-script som läser in kommandon från terminalen och sidan svarar olika beroende på vilket kommando som lästs in. Scriptet ska minst känna till följande kommandon:

- inloggad – Skriv ut användarnamnet på den som är inloggad.
- klockan – Skriv ut vad klockan är (OBS! Endast tiden ska visas).
- datum – Skriv ut dagens datum (OBS! Endast datumet ska visas).
- processer – Skriv ut de processer som användaren kör.
- avsluta – Skriv ut "Hej då samt avsluta scriptet."

Exekveringen av scriptet ska inte avslutas innan användaren skriver in "avsluta".

1. Skriv ett bash-script som utför ovanstående.

## 6 Laborationsfrågor

1. Vilka säkerhetsproblem kan du hitta relaterat till NIS? Hitta ett alternativt sätt att göra samma sak som NIS gör och förklara vilka säkerhetsproblem detta kan lösa eller skapa.