Vaddå Operativsystem?

Ett operativsystem (OS) är den grundläggande programvaran som fungerar som en brygga mellan datorns hårdvara och användaren/applikationerna.

Det är datorns "själ" som gör det möjligt att starta, hantera resurser och köra program.

Utan ett OS är en dator bara en samling av hårdvara.

“Dirigenten”

Hantering av Hårdvara: Styr och kommunicerar med CPU, minne (RAM), lagringsenheter, tangentbord, mus, skrivare m.m.

Körning av Applikationer: Ger en miljö där program kan exekveras.

Filhantering: Organiserar och hanterar data på lagringsenheter (t.ex. hårddiskar).

Minneshantering: Fördelar och allokerar RAM-minne mellan olika program.

Användargränssnitt: Erbjuder ett sätt för användaren att interagera med datorn (t.ex. via en grafisk yta eller kommandorad).

“Delar”

Kärnan (Kernel): Den innersta, centrala delen. Den startar systemet, hanterar resurstilldelningen och kommunicerar direkt med hårdvaran.

Skalet (Shell): Hanterar kommunikationen mellan användaren/applikationer och kärnan. Detta är där användarkommandon tolkas.

Programbibliotek (System Libraries): En samling av gemensamma funktioner och resurser som andra program kan anropa och använda sig av.

“Desktop-OS”

Funktioner: Stöd för avancerad grafik, spel, kontorsprogram, filhantering och multitasking (köra flera program samtidigt).

Ex

Microsoft Windows: Världens mest utbredda OS för PC. Känt för bred kompatibilitet.

macOS (Apple): Känt för sitt eleganta gränssnitt och robusta prestanda, primärt för Apple-hårdvara.

Linux (t.ex. Ubuntu, Fedora): Öppen källkod, mycket flexibelt och säkert. Populärt bland utvecklare och på servrar.

“Mobila OS”

Utvecklade specifikt för smartphones och surfplattor, optimerade för pekskärmar, batteritid och trådlös kommunikation.

Funktioner: Pekgränssnitt, energihantering, platsbaserade tjänster, app-ekosystem.

Exempel:

Android (Google): Linux-baserat och dominerar mobilmarknaden. Känt för flexibilitet och anpassningsbarhet.

iOS (Apple): Operativsystemet för iPhone och iPad. Erbjuder en sömlös och säker upplevelse inom Apples ekosystem.

“Andra OS (server/embedded)”

Server-OS: Hanterar nätverksresurser, webbservrar och databaser. Prioriterar stabilitet, säkerhet och prestanda.

Exempel: Windows Server, Linux-distributioner (t.ex. Red Hat, Debian), Unix.

Inbyggda (Embedded) OS: Låg resursförbrukning, utformade för att köra en enda uppgift i en dedikerad enhet.

Exempel: Används i routrar, mikrovågsugnar, bilar, medicinsk utrustning.

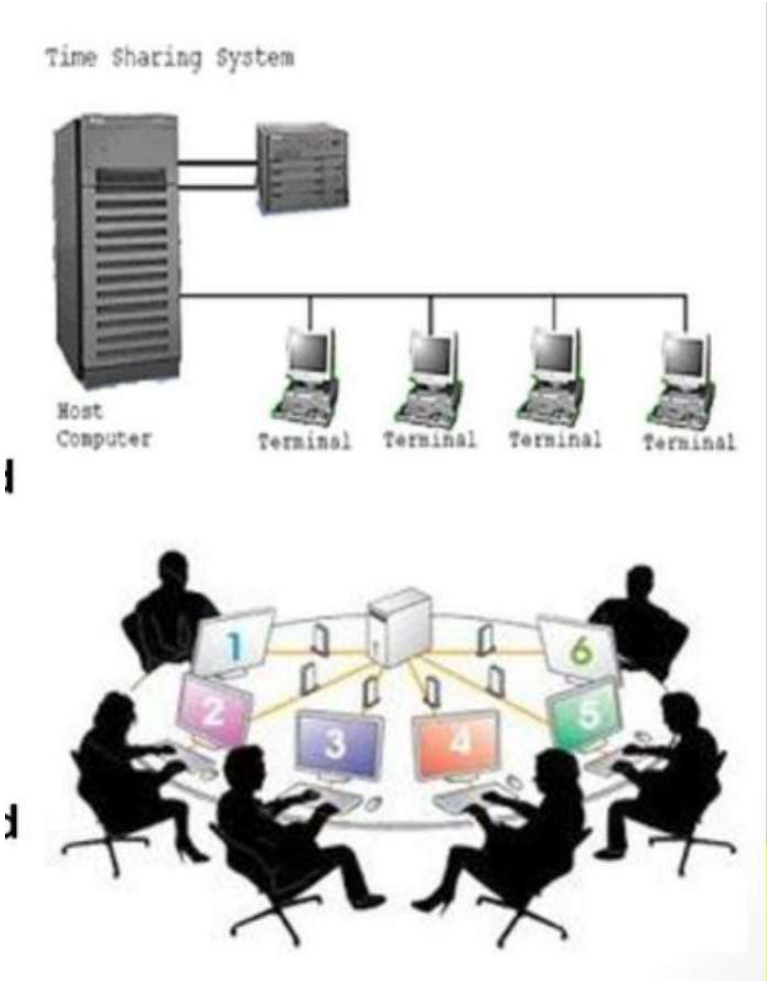
Realtidsoperativsystem (RTOS): Kräver exakt timing och snabba svarstider för kritiska applikationer.

Exempel: Används i robotteknik och industriella styrsystem..

Datorer förr i tiden…

Timesharing allows a central computer to be shared by a large number of users sitting at terminals. Each program in turn is given use of the central processor for a fixed period of time. When the time is up, the program is interrupted and the next program resumes execution. This is called time slicing.

Flera users



https://education.systementor.se

SYSTEMENTOR EDUCATION

Multitasking?

Samma! En dator...fast bara EN user...fast
MÅNGA *samtidiga* program

Problem

Printing (eller tex LCD)

Om flera samtidigt ???

Kö!

BASH

THE BOURNE-AGAIN SHELL

Vaddå BASH?

Bash är det mest utbredda **kommandoradsgränssnittet** (Shell, terminal) i Linux och Unix-liknande operativsystem (t.ex. macOS).

Syfte: Det är ett program som tar emot textkommandon från användaren och instruerar operativsystemet att utföra dem. -

Funktion: Fungerar som en tolk mellan användaren och operativsystemets kärna (Kernel). -

Bash-skript: Kan användas för att skriva skript (små program) för att automatisera uppgifter

WHY: servrar finns inget UI på tex

pwd /cd

PWD – current dir

```
stefan@LAPTOP-FGBC8OV4:~$ pwd
/home/stefan
stefan@LAPTOP-FGBC8OV4:~$
```

CD – change dir

```
stefan@LAPTOP-FGBC8OV4:~$ cd /
stefan@LAPTOP-FGBC8OV4:/$ ls
bin  boot  dev  etc  home  init  lib  lib64  media  mnt
stefan@LAPTOP-FGBC8OV4:/$ cd usr
stefan@LAPTOP-FGBC8OV4:/usr$ ls
bin  games  include  lib  local  sbin  share  src
stefan@LAPTOP-FGBC8OV4:/usr$ d src
d: command not found
stefan@LAPTOP-FGBC8OV4:/usr$ cd src
stefan@LAPTOP-FGBC8OV4:/usr/src$
```

Echo

echo

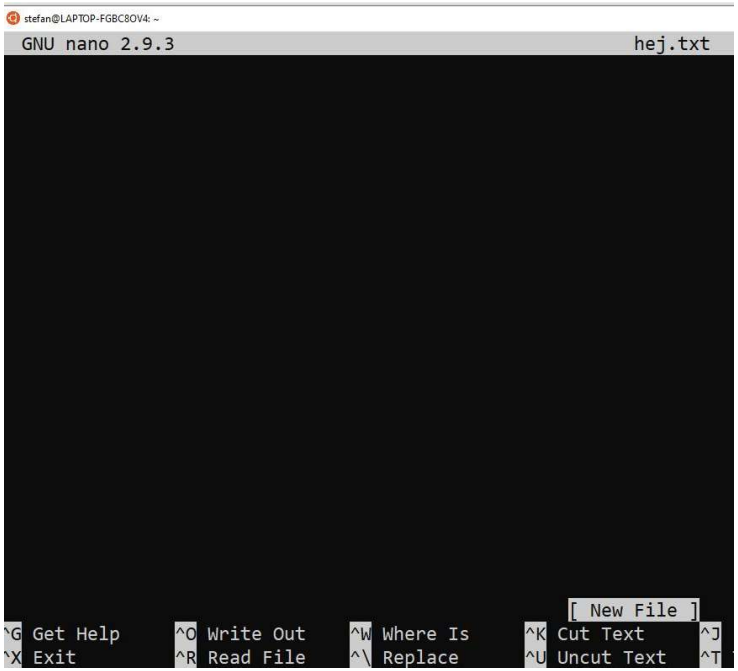
```
stefan@LAPTOP-FGBC8OV4:/usr/src$ echo a text of some kind
a text of some kind
stefan@LAPTOP-FGBC8OV4:/usr/src$ echo $USER
stefan
stefan@LAPTOP-FGBC8OV4:/usr/src$ echo $HOME
/home/stefan
stefan@LAPTOP-FGBC8OV4:/usr/src$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
Apps/CanonicalGroupLimited.UbuntuonWindows_1804.2019.521.0_x64__79rhkp1fndgsc
/mnt/c/WINDOWS/System32/Wbem:/mnt/c/WINDOWS/System32/WindowsPowerShell/v1.0/:
Program Files/dotnet/:/mnt/c/Program Files/Microsoft SQL Server/130/Tools/Bin
rver/Client SDK/ODBC/170/Tools/Binn/:/mnt/c/Program Files/Microsoft SQL Serve
/Program Files (x86)/Microsoft SQL Server/140/Tools/Binn/:/mnt/c/Program File
mnt/c/Program Files/Microsoft SQL Server/140/DTS/Binn/:/mnt/c/Program Files (
/mnt/c/Program Files/Git/cmd:/mnt/c/Users/stefan/AppData/Local/Microsoft/Wind
l/GitHubDesktop/bin:/mnt/c/Users/stefan/AppData/Local/Programs/Microsoft VS C
stefan@LAPTOP-FGBC8OV4:/usr/src$
```

Ex cd $HOME

även om cd ~ är mer använt

Nano!

nano



Create/edit file

Prompten

> to file

>> append to file

TESTA SJÄLV!
echo Hejsan detta är ett meddelande > hej2.txt

cat hej2.txt

Kör echo Hejsan >> hej2.txt många många gånger
Cat ”stannar” inte utan texten scrollar förbi

less hej2.txt

Cat

cat <filnnamn>

less <filnamn>

head <filnamn>

tail <filnamn>

ls

ls -l

```
stefan@LAPTOP-FGBC8OV4:~$ ls -l
total 0
-rw-rw-rw- 1 stefan stefan   22 Nov 10 11:45 hej.txt
-rw-rw-rw- 1 stefan stefan   74 Nov 10 11:47 hej2.txt
drwxrwxrwx 1 stefan stefan 4096 Nov  2 17:04 projects
stefan@LAPTOP-FGBC8OV4:~$
```

If the first character is a `-` the item is a file, if it is a `d` the item is a directory. The rest of the string is three sets of three characters. From the left, the first three represent the file permissions of the *owner*, the middle three represent the file permissions of the *group* and the rightmost three characters represent the permissions for *others*. In each set, an `r` stands for read, a `w` stands for write, and an `x` stands for execute.

If the `r`, `w`, or `x` character is present that file permission is granted. If the letter is not present and a `-` appears instead, that file permission is not granted.

Skapa och hantera folder

mkdir <namn>

rmdir <namn>

mkdir -p hello/there/test

rm -r <folder>

Leta i filer

grep <pattern> <filer>

-i case insensitive
-r rekursivt I kataloger

Curl

```
sudo apt-get install curl
```

Suppose you want to retrieve a single file from a C

```
stefan@LAPTOP-FGBC8OV4:~$ curl http://www.nackademin.se
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://www.nackademin.se/">here</a>.</p>
</body></html>
stefan@LAPTOP-FGBC8OV4:~$
```

-o för utfil

Mer

Ps (ps -e) ps -e | grep bash

mv

```
stefan@LAPTOP-FGBC8OV4:~/projects$ echo Hello World
Hello World
stefan@LAPTOP-FGBC8OV4:~/projects$ mkdir bashscripts
stefan@LAPTOP-FGBC8OV4:~/projects$ cd bashscripts
stefan@LAPTOP-FGBC8OV4:~/projects/bashscripts$ nano first.sh
stefan@LAPTOP-FGBC8OV4:~/projects/bashscripts$ first

Command 'first' not found, but can be installed with:

sudo apt install yagiuda

stefan@LAPTOP-FGBC8OV4:~/projects/bashscripts$ first.sh
first.sh: command not found
stefan@LAPTOP-FGBC8OV4:~/projects/bashscripts$ ./first.sh
-bash: ./first.sh: Permission denied
stefan@LAPTOP-FGBC8OV4:~/projects/bashscripts$ chmod 755 first.sh
stefan@LAPTOP-FGBC8OV4:~/projects/bashscripts$ ./first.sh
Hello World
stefan@LAPTOP-FGBC8OV4:~/projects/bashscripts$
```

Variabler, tilldelning

```
city=New_York
namn="Stefan Holmberg"
age=47
email = 'stefan@whatever.com'
ageLastYear = age-1

echo $city
echo $namn
echo $age
echo $email
echo $ageLastYear
```

Email ??? mellanslag

AgeLastYear??? - age-1 ????

Enda som finns är...STRÄNGAR!

```
stefan@LAPTOP-FGBC8OV4:~$ ageLastYear=age-1
stefan@LAPTOP-FGBC8OV4:~$ echo  $ageLastYear
age-1
stefan@LAPTOP-FGBC8OV4:~$
```

LET

```
stefan@LAPTOP-FGBC8OV4:~$ let ageLastYear=age-1
stefan@LAPTOP-FGBC8OV4:~$ echo $ageLastYear
46
```

String concat

```
7  #
8  FIRST_NAME='Stefan'
9  LAST_NAME='Holmberg'
10 echo $FIRST_NAME$LAST_NAME
11 FULL_NAME=$FIRST_NAME$LAST_NAME
12 echo $FULL_NAME
```

```
StefanHolmberg
StefanHolmberg
```

Allt är ju strängar...

```
7  #
8  tal1=1
9  tal1+=2          # 1+2 => 12
10 tal1+=$tal1    # 12+12 => 1212
11 echo $tal1
```

```
1212
```

Remember...

```
#
tal1=1
tal1=tal1+2
echo $tal1
```

```
tal1+2
```

Så att plussa
blir med LET

```
tal1=1
let tal1=tal1+2
echo $tal1
```

```
3
```

Operators...

- `+` add
- `-` subtract
- `*` multiply
- `/` divide
- `%` modulo (remainder of the division)
- `**` exponentiation

Logiska...

- `&&` logical AND
- `||` logical OR

Comparison

- `<`
- `<=`
- `==`
- `>=`
- `>`

- `-lt` lower than
- `-gt` greater than
- `-le` lower or equal than
- `-ge` greater or equal than
- `-eq` equal to
- `-ne` not equal to

IF

Simple `if`:

```
if condition
then
  command
fi
```

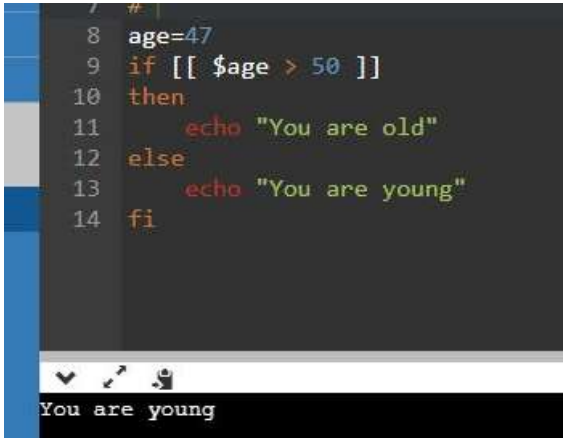
`if` then `else`:

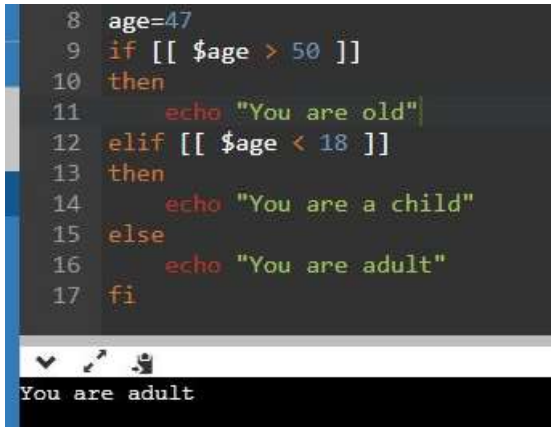
```
if condition
then
  command
else
  anothercommand
fi
```

Nested `if` - then - `else`:

```
if condition
then
  command
elif
  anothercommand
else
  yetanothercommand
fi
```

IF - exempel

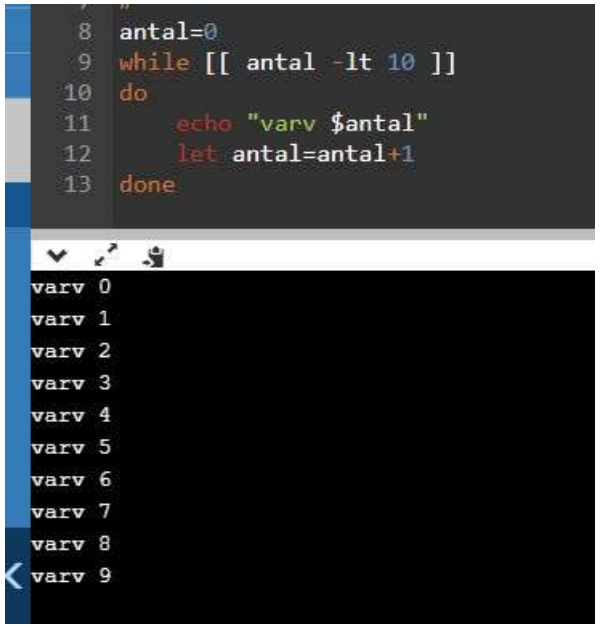




AND/OR

```
if [[ $age == 47   ]] && [[ namn -eq "Stefan" ]]
if [[ $age == 47   ]] || [[ namn -eq "Stefan" ]]
```

Loops


8 antal=0
9 while [[antal -lt 10]]
10 do
11 echo "varv $antal"
12 let antal=antal+1
13 done
varv 0
varv 1
varv 2
varv 3
varv 4
varv 5
varv 6
varv 7
varv 8
varv 9

List – for loops

```
8  stefansBarn="Fanny Josefine Oliver"
9  for namn in $stefansBarn
10 do
11     echo $namn
12 done
```

```
Fanny
Josefine
Oliver
```

Auto list space separated

Deklarera en array

```
8  declare -a StringArray=("Linux Mint" "Fedora" "Red Hat Linux" "Ubuntu" "Debian" )
9  for val in ${StringArray[@]}
10 do
11    echo $val
12 done
```

List – for loops

Iterate with *

```
8  LanguageArray=("PHP"  "Java"  "C#"  "C++"  "VB.Net"  "Python"  "Perl")
9  for val1 in ${LanguageArray[*]}
10 do
11     echo $val1
12 done
```

Case

```
#!/bin/bash
read -p "How many shoes do you have?" value
case $value in
  0|1)
    echo "Not enough shoes! You can't walk"
    ;;
  2)
    echo "Awesome! Go walk!"
    #...
    ;;
  *)
    echo "You got more shoes than you need"
    #...
    ;;
esac
```

Looping over files

```
#!/bin/bash
FILES=/path/to/*.txt
for f in $FILES
do
  echo "Processing $f file..."
  # count number of lines and output that for file $f
  wc -l $f
done
```