



Datorer och Datoranvändning

Föreläsning 1 - Unix, Linux och terminalen

Ask me anything

12 questions
10 upvotes



Föreläsningens innehåll

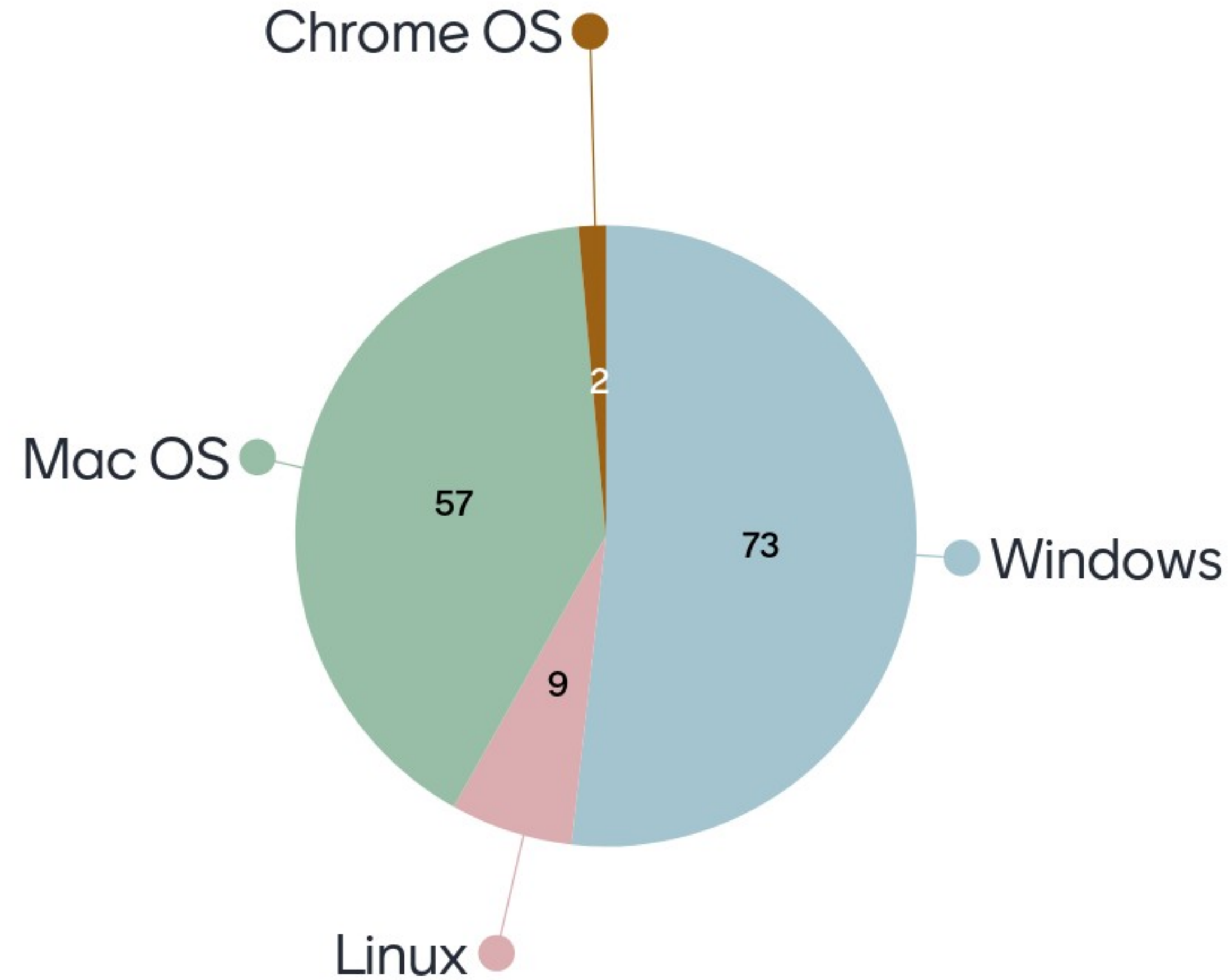
- Operativsystem, Unix-historik
- Filer och kataloger
- Kommandotolken, prova själva ($\backslash(^_^)/$)
 - Kommandon
 - Filhantering
 - Filskydd
 - bash-finesser
- Annat, i mån av tid
 - Vad är ett kommando?
 - Processer.
 - Fönstersystem.
 - Mer om filskydd.



Operativsystem

- En samling program som:
 - möjliggör körning av "vanliga program"
- Hanterar:
 - Alla körande program, "parallellt"
 - Yttre enheter (mus, tangentbord, nätverk, etc.)
 - Lagring av data (filer, ...)
 - Filskydd, felhantering, kommunikation med användaren
- Exempel:
 - Linux, Windows, Mac OS X, Unix, ...

Vilket operativsystem använder du?





Unix historia

- **1969:** Unix utvecklas på AT&T's Bell Labs av Ken Thompson och Dennis Ritchie.
- **1973:** Unix omskrivs i C-språket av Dennis Ritchie för portabilitet.
- **1983:** GNU-projektet startar för att skapa ett fritt Unix-liknande system.
- **1991:** Linux-kärnan skapas av Linus Torvalds, integreras med GNU-program.
- **Idag:** Unix-principer fortsätter påverka moderna operativsystem och mjukvaruutveckling.



Kännetecknen för Unix

- Liten kärna, grundläggande funktioner
- Fleranvändarsystem
- Filskydd
- Kommandotolk (terminal)
- Ursprungligen inte för nybörjare...
 - Men lättare nu med grafiska skrivbordsmiljöer



Filer i Unix

I en dator måste man kunna lagra **program** (Word, Excel, Java-/Scalakompilator, spelprogram, ...) och **data** (brev, hemsidor, programmeringsuppgifter, ...).

Man lagrar program och data i filer. En fil finns normalt på skivminne och har ett namn. Filnamn i Unix/Linux:

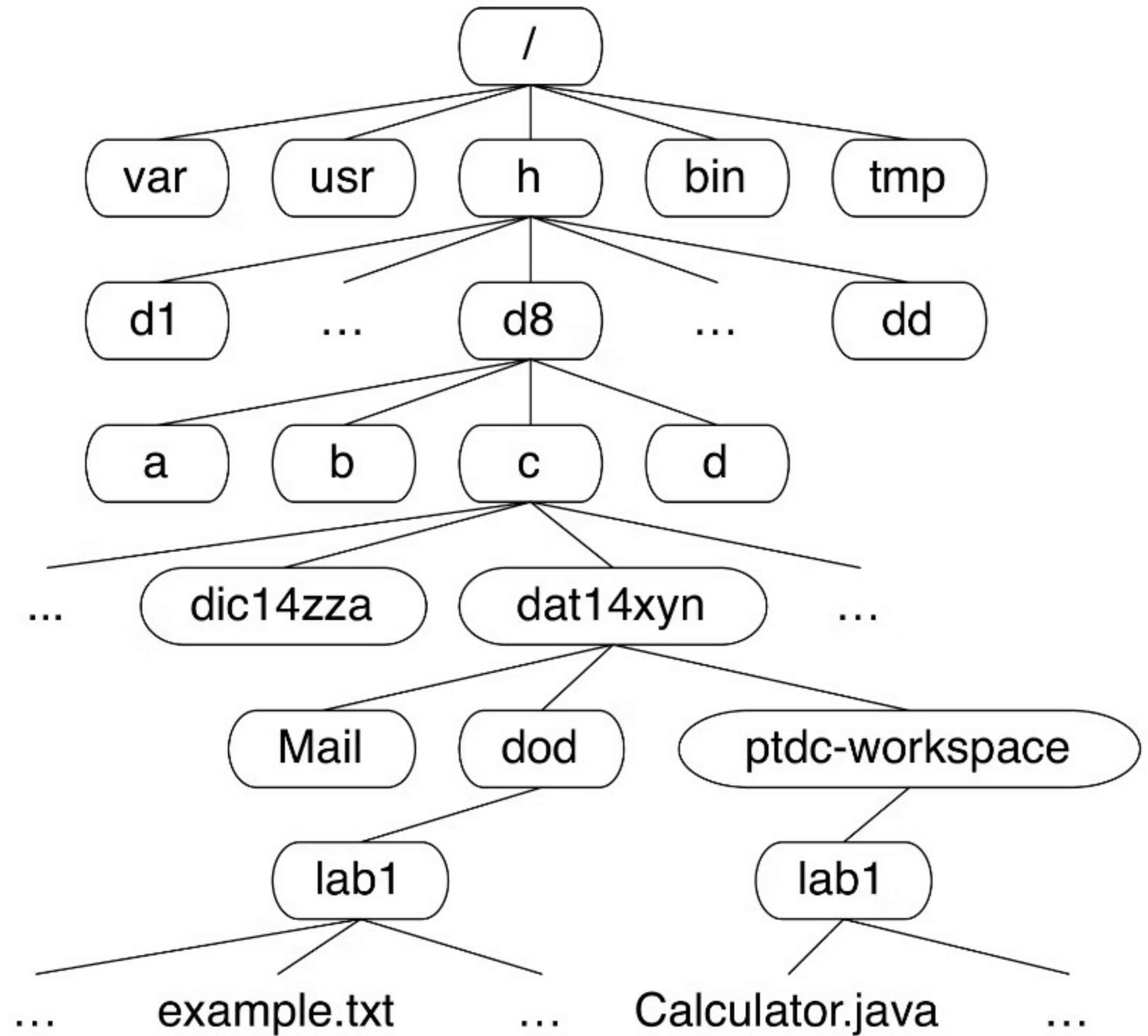
namn.tillägg

(Calculator.java, Calculator.class, brev.txt, test1, ...)

Tillägget är bara en extra upplysning, bestämmer inte filtypen.

Kataloger

Varje fil finns i en katalog (eller mapp).
 Kataloger kan innehålla underkataloger, så
 man får en hierarkisk struktur; ett **filträd**.



Kataloger

För att referera till filer anger man deras namn. Ett fullständigt (absolut) filnamn börjar från roten, och räknar upp alla kataloger med / emellan:

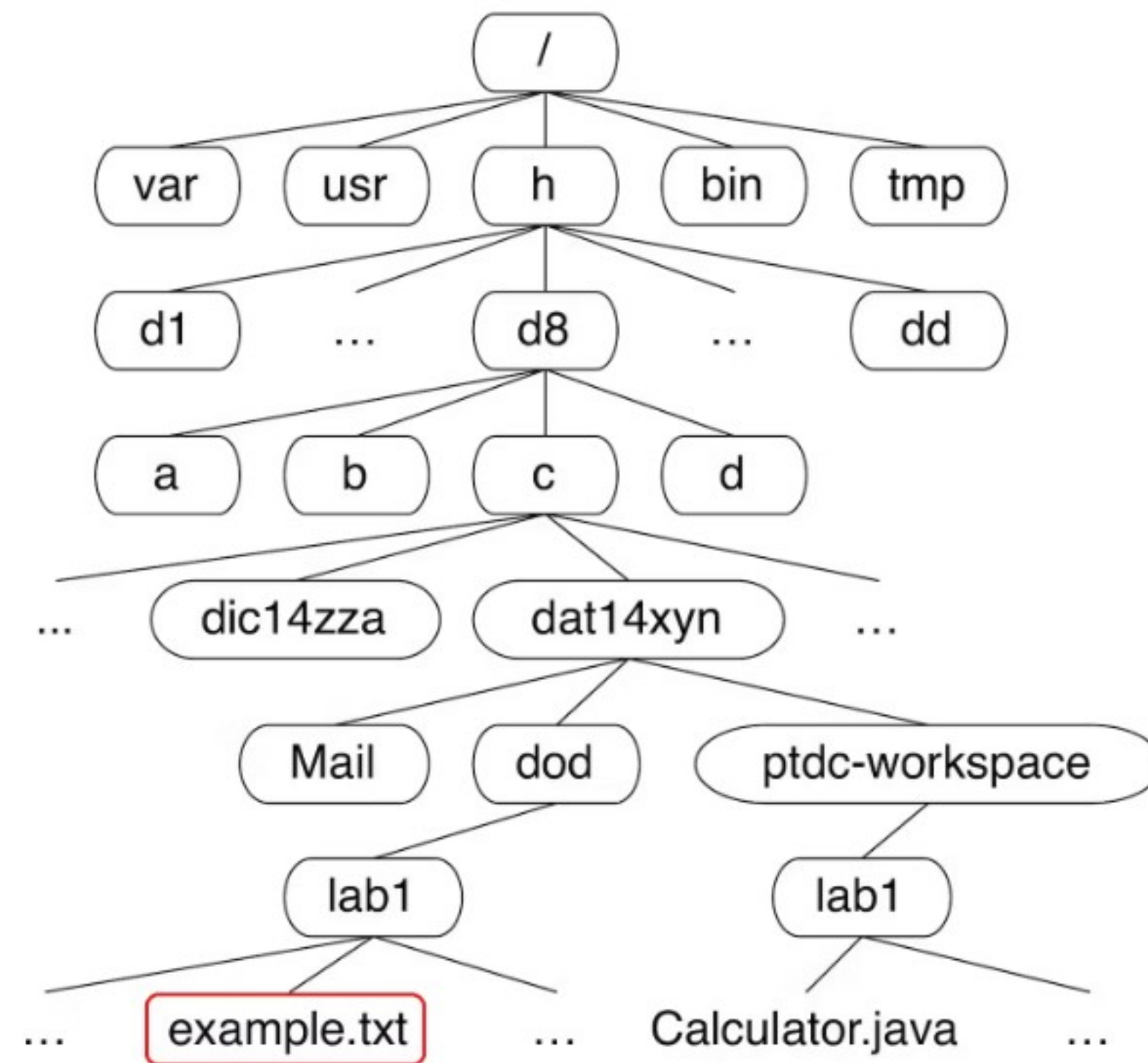
`/h/d8/c/dat14xyn/dod/lab1/example.txt`

Men Unix håller reda på en **aktuell katalog**, som man själv kan ändra. Om **lab1** är aktuell katalog så kan man komma åt ovanstående fil med följande **relativa** namn:

`example.txt`

Om **dat14xyn** är aktuell katalog så kommer man åt filen med namnet:

`dod/lab1/example.txt`



En del filer har speciella, förkortade namn:

~ (tilde): hemkatalogen för aktuell användare

(Tilde skrivs med **Alt Gr + ~**)

~*user*: hemkatalogen för användaren *user*

.(punkt): aktuell katalog

..(punktpunkt): katalogen ovanför aktuell katalog i filträdet

Antag att ~/dod/lab1 är aktuell katalog (blå pil). Man kan navigera i filträdet enligt följande exempel:

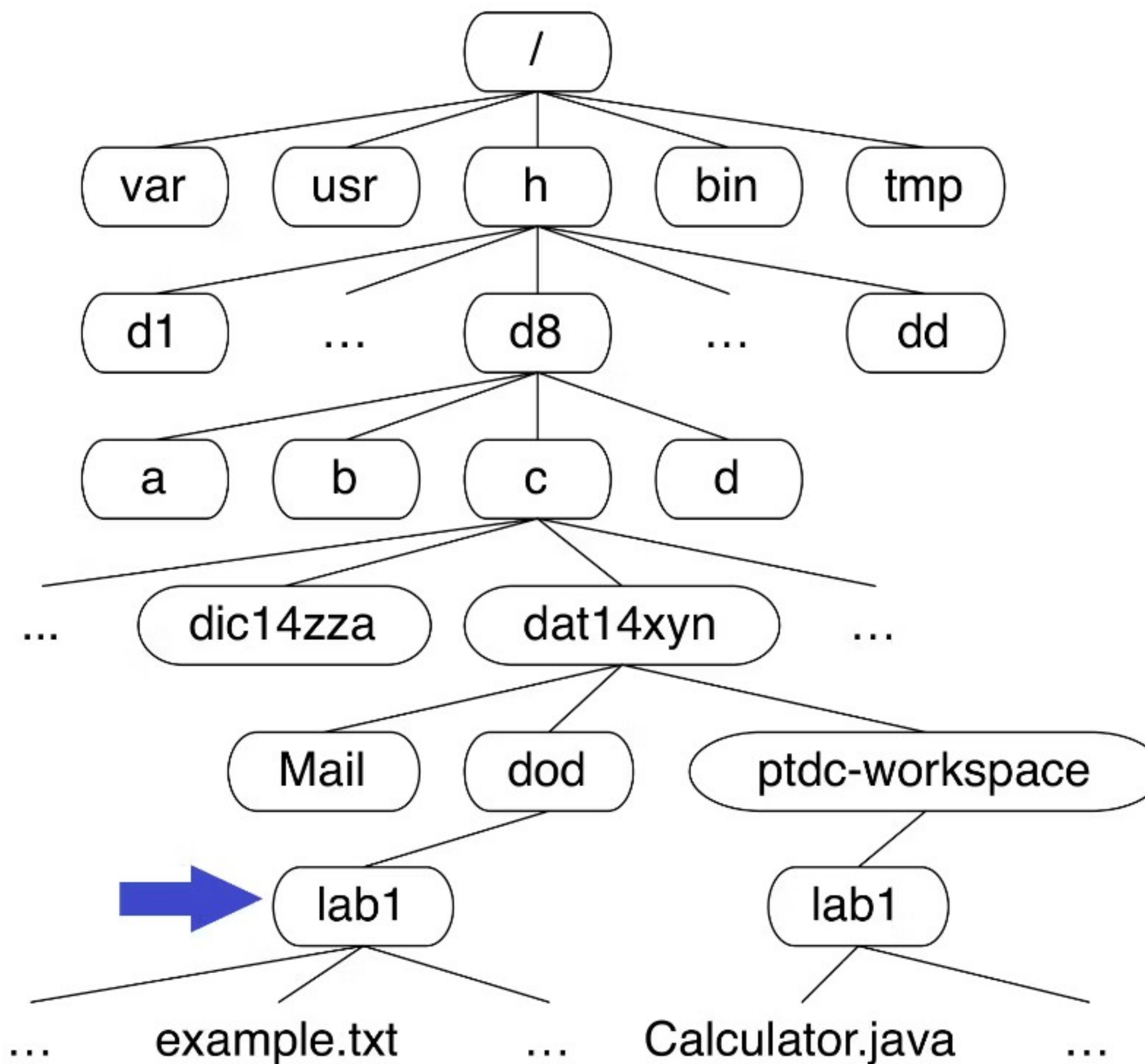
~ — hemkatalogen

. — aktuell katalog

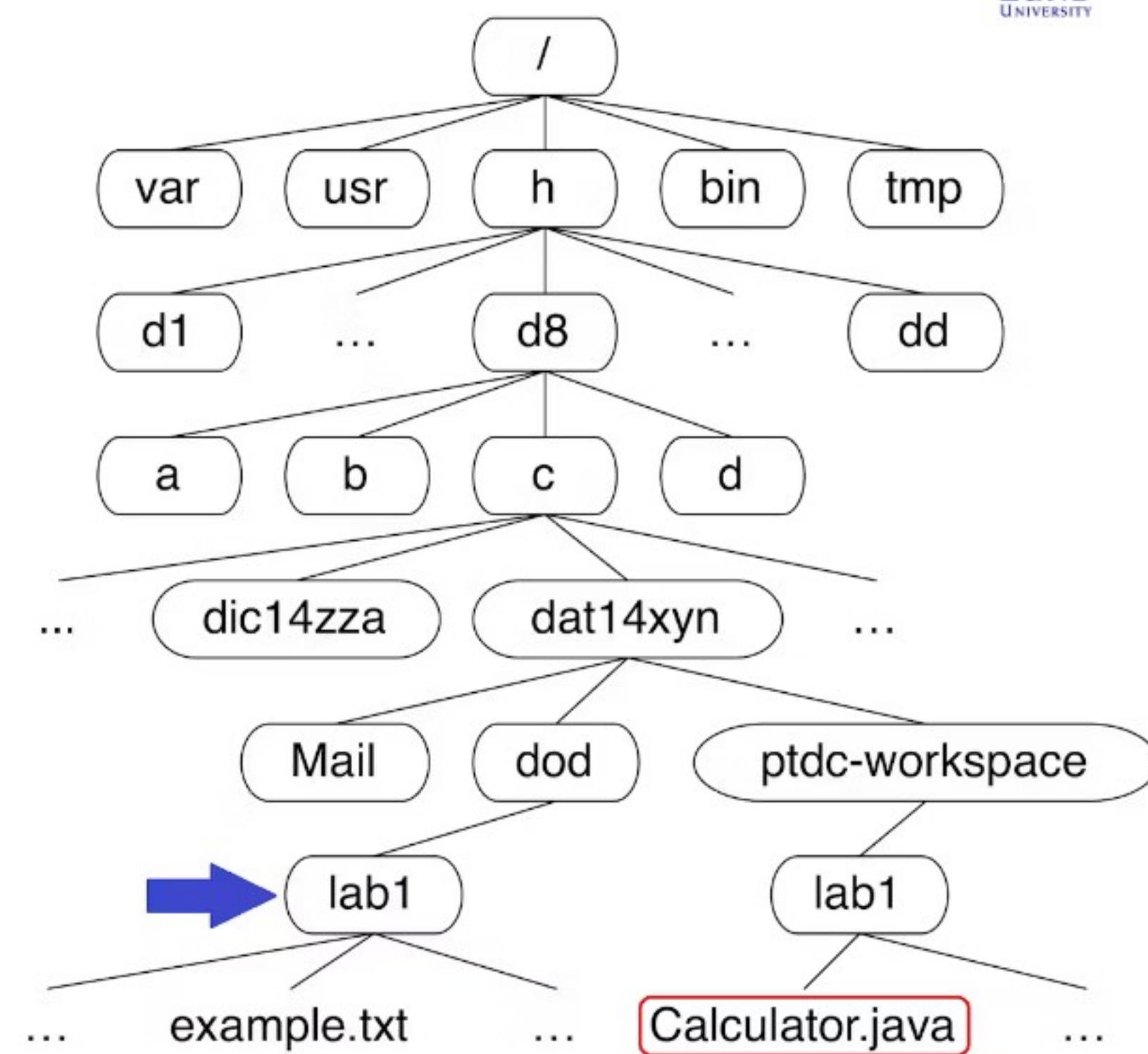
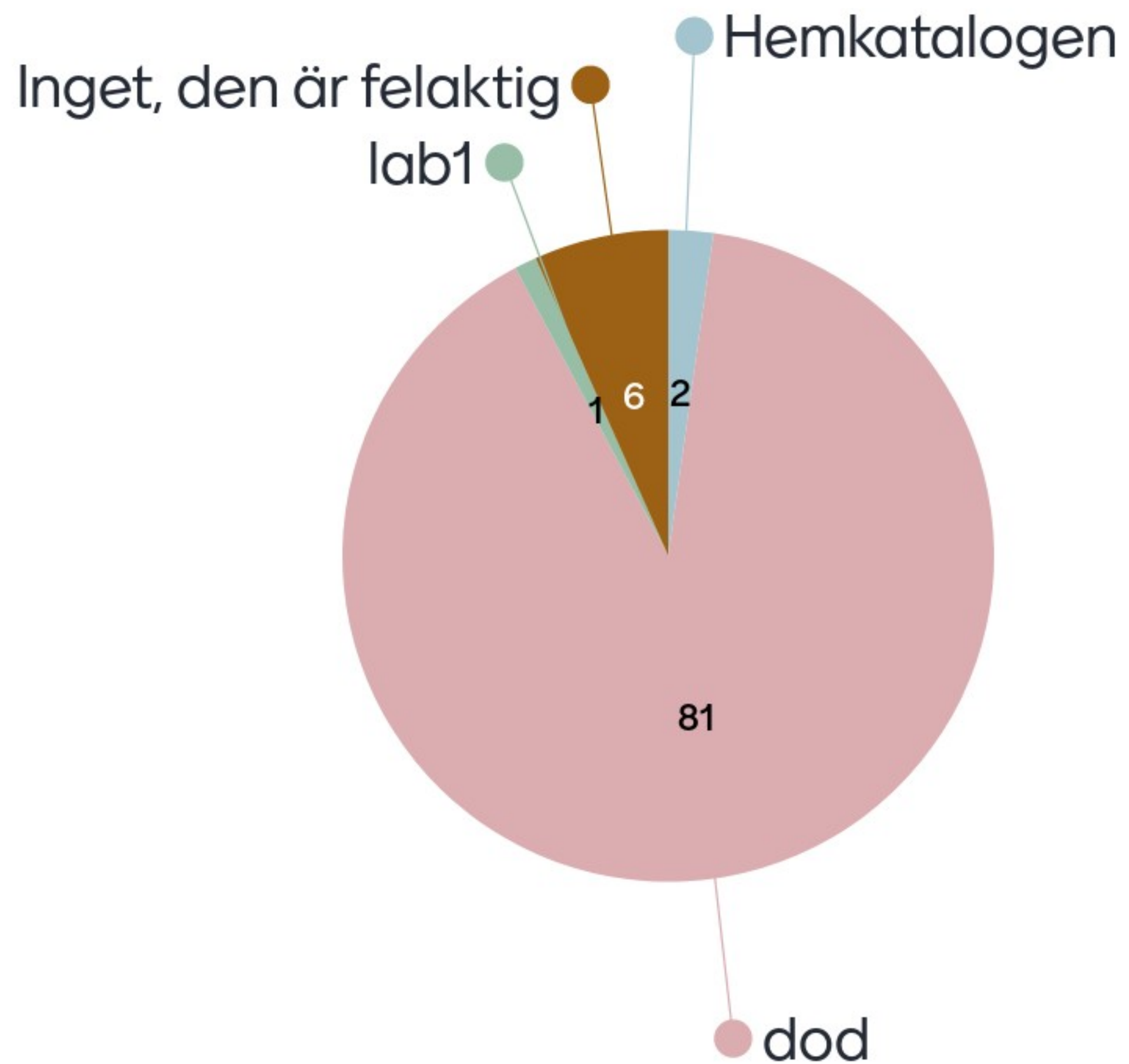
.. — dod-katalogen

./.. — hemkatalogen

./././././ — ???



~/dod/.!./!./!./!./!./dod/.



- ~ (tilde): hemkatalogen för aktuell användare
- ~user: hemkatalogen för användaren user
- . (punkt): aktuell katalog
- .. (punktpunkt): katalogen ovanför aktuell katalog

Navigera i filträdet

29 Responses

~/ptdc-
workspace/lab1/Calculator.java

33 votes

~/ptdc-
workspace/lab1/Calculator.java

16 votes

../..ptdc-
workspace/lab1/Calculator.java

13 votes

~/ptdc-
workspace/lab1/calculator.java

6 votes

../..ptdc-
workspace/lab1/Calculator.java

5 votes

~/ptdc-
workspace/lab1/Calculator.java

5 votes

~/calculator.java

3 votes

~/ptdc-
workspace/lab1/Calculator.java

3 votes

~/ptdc-
workspace/lab1/Calculator.java

3 votes

Navigera i filträdet

29 Responses

```
~/ptdc-  
workspace/lab1/calculator.java
```

2 votes

```
../ptcd-  
workspace/lab1/Calculator.java
```

2 votes

```
/h/d8/c/dat14xyn/ptdc-  
workspace/lab1/Calculator.java
```

1 vote

```
~/ptdc-  
workspace/lab1/Calculator.java
```

2 votes

```
lab1/Calculator.java
```

1 vote

```
find / -iname "Calculator"
```

1 vote

```
find / -iname "Calculator"
```

2 votes

```
../calculator.java
```

1 vote

```
~/ptdc-workspace/lab/calculator.java
```

1 vote

Navigera i filträdet

29 Responses

```
../ptdc-  
workspace/lab1/Calculator.java
```

1 vote

```
../ptdc-  
workspace/lab1/Calculator.java
```

1 vote

```
find / -iname "Calculator"
```

1 vote

```
~/ptdc-  
workspace/lab1/Calculator.java
```

1 vote

```
~/ptdc-  
workspace/lab1/calculator.java
```

1 vote

```
~/ptdc-workspace/lab1
```

```
~/ptdc-  
workspace/lab1/Calculator.java
```

```
~/ptdc-  
workspace/lab1/calculator.java
```

```
~/ptdc-  
workspace/lab1/Calculator.java
```

Navigera i filträdet

29 Responses

```
~/ptdc-  
workspace/lab1/Calculator.java
```

```
~/ptdc-  
workspace/lab1/Calculator.java
```


Efter pausen: Kommandotolken

12 questions
10 upvotes

Kommandotolken

Man kommunicerar med Unix genom att ge kommandon. Ett särskilt program i operativsystemet, **kommandotolken**, eller "**shell**", läser in kommandon och utför dem.

Det finns flera olika kommandotolkar. På Linuxdatorerna i E-huset används kommandotolken **bash** ("Bourne-Again SHell"). Andra vanliga kommandotolkar är sh, zsh och csh.

Kommandotolken läser kommandon som ges i ett terminalfönster på skärmen. Så här arbetar tolken:

Upprepa i all oändlighet:

Läs ett kommando

Om kommandot är ett riktigt kommando:

Utför det som ska göras

Annars:

Skriv ut "XXX: command not found"



Kommandon

- Varje kommando skrivs på en rad:
 - kommandonamn -option1 -option2 ... argument1 ...
- Kommandot talar om vad som ska göras.
- Argumenten är (oftast) filer eller kataloger som påverkas av kommandot.
- Optionerna modifierar kommandot på något sätt.

Prova själv!

- Vad gör följande kommandon?
 - pwd
 - ls
 - cd

Filhantering

- mkdir
- touch
- Editera filen (nano, code, notepad, ...)
- less
- cp
- mv
- rm

Filskydd

Varje fil har en ägare. Ägaren identifieras med användarnamn och grupp, till exempel **dat14xyn** i gruppen **students**.

Ägaren kan skydda filer så att andra inte kan komma åt dem eller öppna dem.

Filskydd och ägare (och storlek och datum) skrivs ut om man gör kommandot: **ls -l**

```
hacke-3{dat14xyn}: ls -l
-rw-r----- 1 dat14xyn students 4940 aug 21 11:14 example.txt
  skydd      ägare    grupp
```



bash-finesser (1/3)

Man kan editera den aktuella kommandoraden:

Backspace/Delete – Ta bort tecknet till vänster/höger om markören.

← → – Flytta markören på raden.

Tab – Filnamnskomplettering

Control-U – Radera allt till vänster om markören.

Control-K – Radera allt till höger om markören.

Control-L – Töm terminalfönstret.



bash-finesser (2/3)

bash håller reda på de senaste kommandona som utförts, och man kan få tillbaka gamla kommandon:

↑ ↓ — Bläddra bland tidigare kommandon.

!! — Gör om det senaste kommandot.

!**abc** — Gör om det senaste kommandot som inleds med *abc*, till exempel !javac



bash-finesser (3/3)

När man refererar till filer kan man utnyttja jokertecken (wildcards):

? — Motsvarar *ett* godtyckligt tecken.

* — Motsvarar *noll eller flera* godtyckliga tecken.

Exempel, vad gör dessa kommandon?

```
javac Test?.java
```

```
cp *.txt backup/
```

Quiz!

11 Responses

```
rm *.class
```

```
*.class
```

```
rm *.class
```

```
rm *.class
```

```
class begone
```

```
rm -rf *.class
```

```
rm *.class
```

```
rm *.class
```

```
rm *.class
```

Quiz!

11 Responses

rm *.class

rm *.class

Antag att vi har följande
filer i vår katalog:

Bank.class
Bank.java
FinalReport.tex
Person.class
Person.java
PersonBankIntegrateTest.class
PersonBankIntegrateTest.java
TestBank.class
TestBank.java
TestPerson.class
TestPerson.java
fig1.png
fig2.png
fig2_cropped.png
outline.tex
todo-list.txt
figures/
test/

Quiz!

Vad är egentligen ett kommando?

Kommandon är av två slag:

“inbyggda” — små kommandon som exekveras direkt i kommandotolken, till exempel `cd`, `pwd`, `exit`

“vanliga program” — kommandotolken letar efter ett program med samma namn som kommandot

Kommandotolken letar efter program i de kataloger som anges i sökvägen, **PATH**. Det är en operativsystemvariabel.

*(PATH kan uppdateras med kommandot **export**).*

När man till exempel skriver `javac X.java`, så exekveras programmet `/usr/bin/javac`, eftersom katalogen `/usr/bin` finns i sökvägen.

När man ska exekvera ett eget program som finns i den aktuella katalogen skriver man:

`./programnamn`

Inledande `./` är nödvändigt eftersom `.` (den aktuella katalogen) inte finns i sökvägen.





Processer

Varje program som körs i Unix körs som en egen process som exekverar självständigt, "parallellt" med andra processer. I själva verket kan processorkärnan bara göra en sak i taget. Varje process körs några milli- eller microsekunder, sedan får nästa process exekvera, osv. Detta hanteras av operativsystemet.

När man exekverar ett program från ett terminalfönster "låser" programmet fönstret tills det avslutas – så vill man ofta att det ska fungera.

Men man kan också köra program "i bakgrunden" dvs frikopplade från terminalfönstret. Det gör man genom att skriva `&` sist på kommandoraden, till exempel:

```
gedit example.txt &
```



Fönstersystem, skrivbordsmiljö

I Unix och Linux är fönstersystemet inte inbyggt (Unix utvecklades långt innan det fanns grafiska skärmar).

De flesta Unix- och Linuxsystem som kör på persondatorer använder fönstersystemet **X Window System**, som oftast kallas bara **X**. **X** gör det möjligt att ha fönster på skärmen, att flytta dem, ikonifiera dem, osv.

Exakt hur fönstren ska se ut och hur man arbetar med dem bestäms av fönsterhanteraren, som också är ett separat program. Ovanpå fönsterhanteraren finns skrivbordsmiljön. Det bästa (enda) sättet att lära sig att använda en skrivbordsmiljö är att träna!

Mer om filskydd (1/2)

Filskyddet visas när man gör `ls -l`, men vad betyder det egentligen? Se till höger →

Det första tecknet är alltid:

d — För kataloger (directory)

- — För vanliga filer

```
-rw-r----- 1 dat14xyn students 4940 aug 21 11:14 example.txt
  u  g  o
```

Tre kategorier av användare:

- u** (user) filens ägare
- g** (group) medlemmar i samma grupp som ägaren
- o** (others) alla andra

Tre olika rättigheter:

- r** (read) tillstånd att läsa
- w** (write) tillstånd att skriva
- x** (execute) tillstånd att exekvera program (för kataloger betyder **x** tillstånd att titta på innehållet i katalogen)

Mer om filskydd (2/2)

Kommando för att ändra filskydd:

```
chmod skydd fil1 fil2 ...
```

Filskydd kan anges symboliskt, till exempel:

```
u=rw,o=r
```

Men också numeriskt, där man anger skydden med ettor (påslagna) eller nollor (avslagna). Till exempel:

```
r = 100, w = 010, x = 001
```

```
rx = 101, rw = 110, rwx = 111
```

Siffrorna tolkar man sedan som binära tal, vilket ger:

```
x = 1, w = 2, r = 4, rx = 5, rw = 6, rwx = 7
```

Fullständiga numeriska filskydd blir till exempel:

```
604 <=> u=rw,o=r
```

```
705 <=> u=rwx,o=rx
```

