



Föreläsning II

DVA249/DVA267

Linux

Peter Backeman

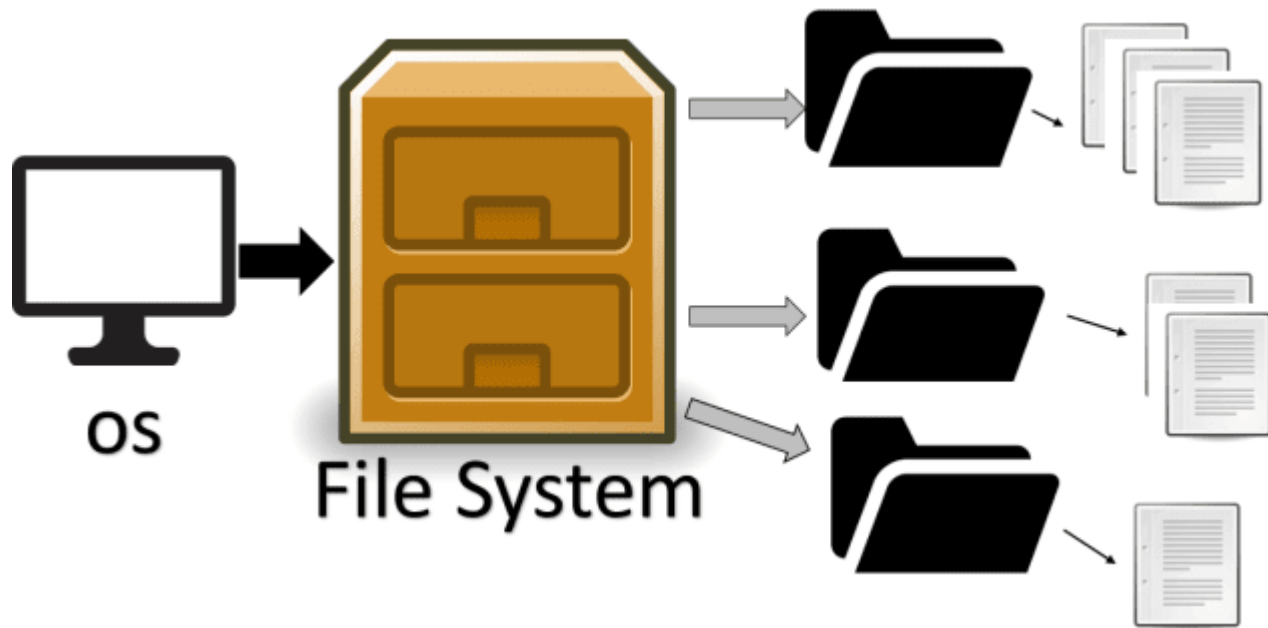
IDT/NetCenter

DVA249/DVA267 HT2023



Föreläsningens innehåll

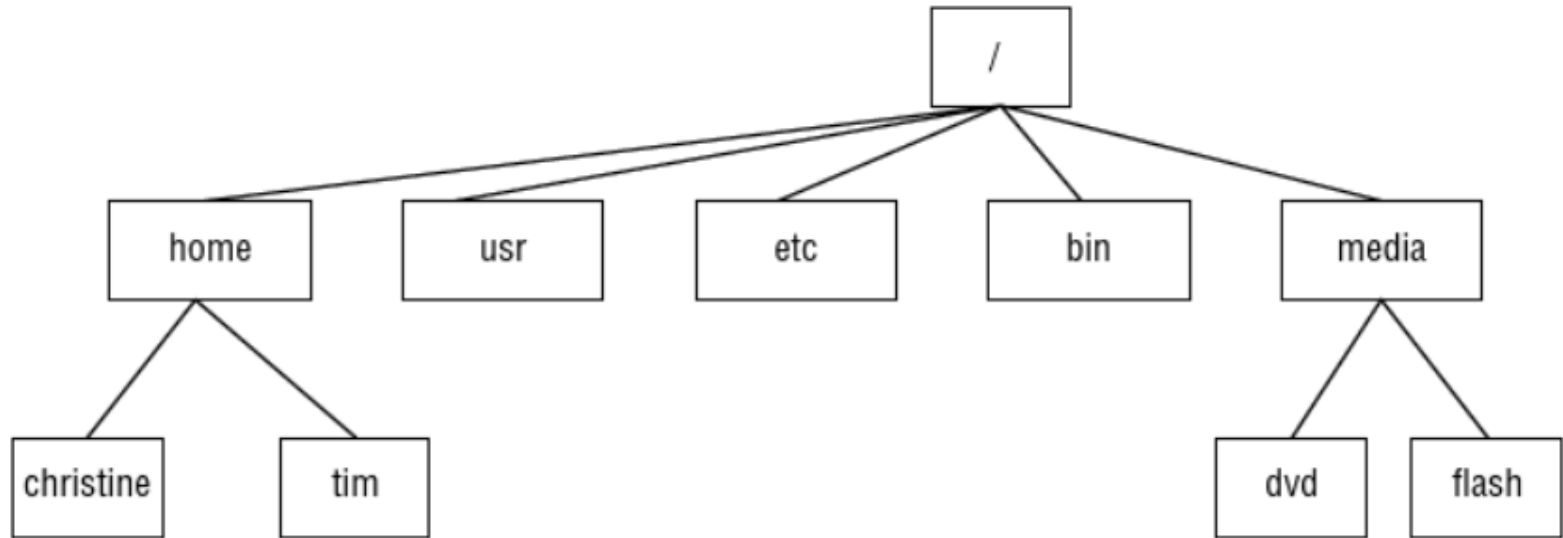
- Tar upp mycket av det som står i litteraturen
- Täcker inte allt
- Workshop på fredag för repetition och förtydligande



Vad finns det för filer?

- Program
- Ikoner, typsnitt
- Konfigurationsfiler
- Databaser
- Logfiler
- ...
- Användarfiler

FIGURE 7.1 In Linux, all files are referred to relative to a single root (/) directory.



Guidad Visning

Katalogstruktur

TABLE 7.2 Important Linux directories according to the FHS

Directory	Purpose
/	The root directory. All files appear in this directory or subdirectories of it.
/etc	Holds system configuration files
/boot	Holds important boot files, such as the Linux kernel, the initial RAM disk, and often boot loader configuration files
/bin	Holds program files that are critical for normal operation and that ordinary users may run
/sbin	Holds program files that are critical for normal operation and that ordinary users seldom run
/lib	Holds libraries—code used by many other programs—that are critical for basic system operation
/usr	Holds programs and data used in normal system operation but that are not critical for a bare-bones boot of the system. This directory is split into subdirectories that mirror parts of the root organization— <code>/usr/bin</code> , <code>/usr/sbin</code> , <code>/usr/lib</code> , and so on.

/home	Holds users' home directories. Separating this directory into its own low-level filesystem effectively isolates most user data from the OS, which can be useful if you want to reinstall the OS without losing user data.
/root	The root user's home directory
/var	Holds miscellaneous transient files, such as log files and print spool files. One subdirectory of <code>/var</code> , <code>/var/tmp</code> , deserves special mention. Like <code>/tmp</code> (described next), <code>/var/tmp</code> holds temporary files. These files should <i>not</i> be deleted when the computer reboots.
/tmp	Holds temporary files, often including temporary files created by user programs. Such files may theoretically be deleted when the computer reboots, although in practice many distributions don't do this.
/mnt	The traditional mount point for removable media; sometimes split into subdirectories for each mounted filesystem
/media	The new mount point for removable media; typically split into subdirectories for each mounted filesystem
/dev	Holds device files, which provide low-level access to hardware
/run	Information about the running system

Kataloghantering

Kataloghantering

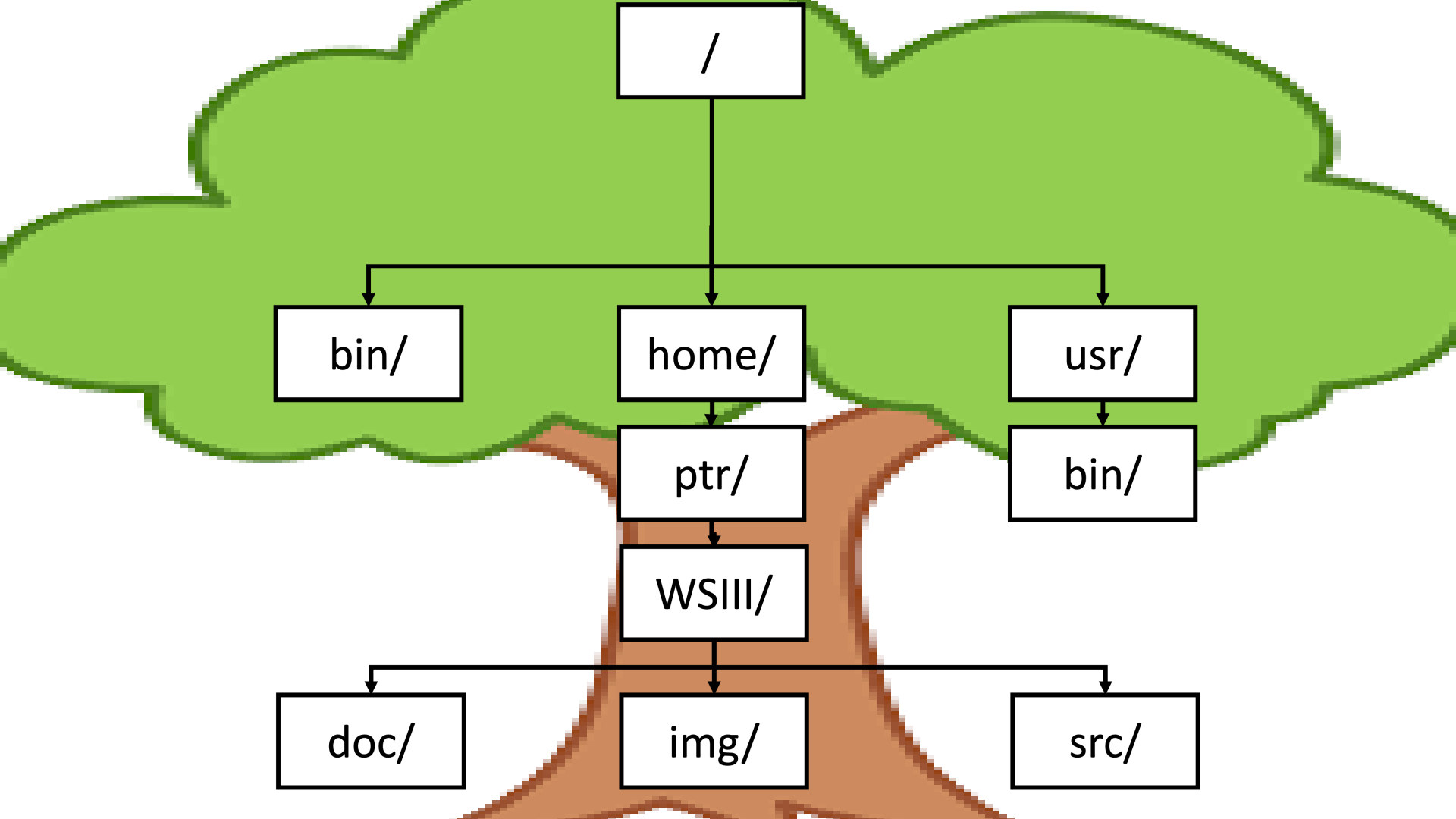
- Skapa en katalog: `mkdir`
 - Var hamnar katalogen?
 - Vad heter katalogen?
- Ta bort katalog : `rmdir`
 - Vilken katalog tas bort?

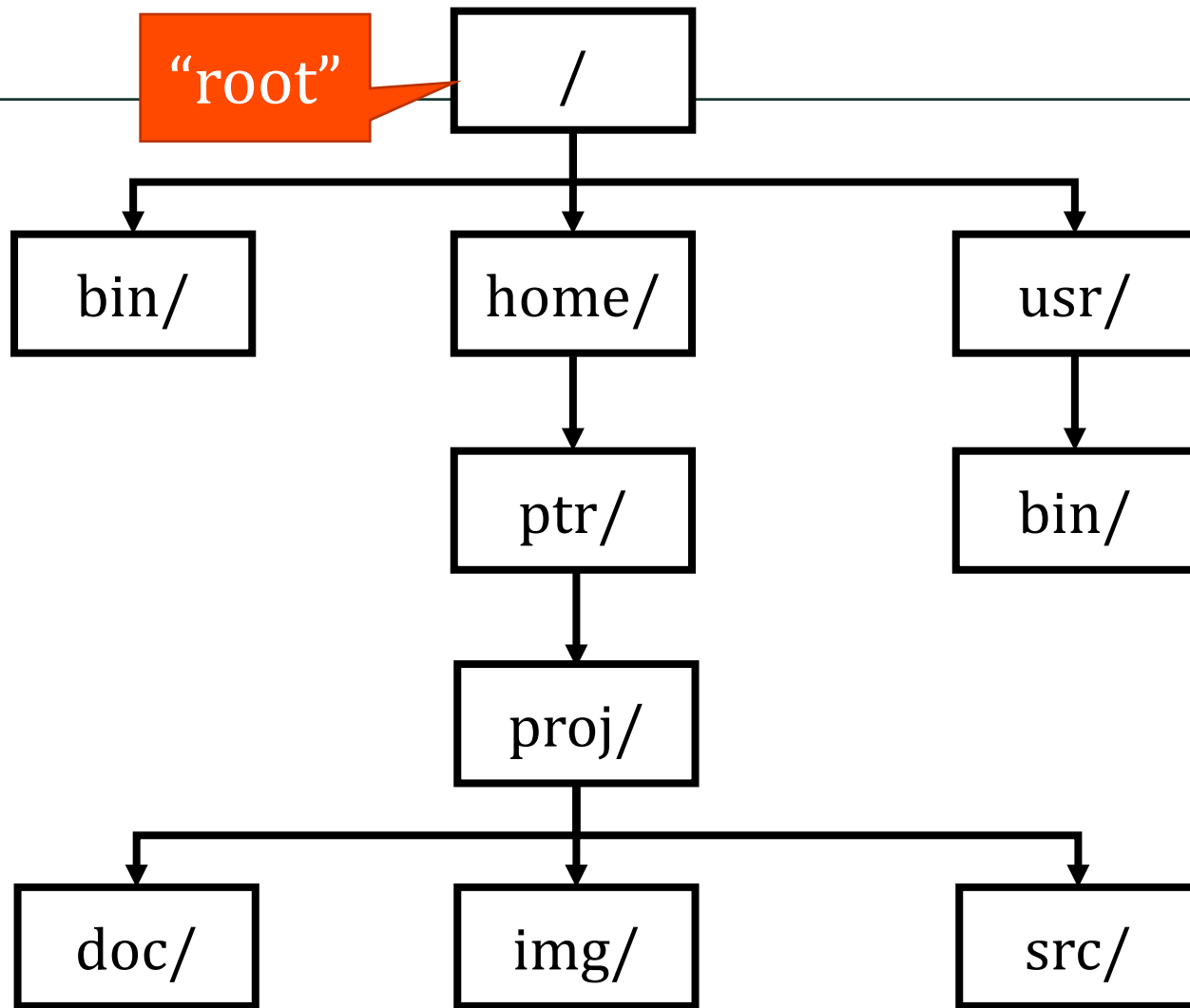
A photograph of a forest path in autumn. The path is a narrow, winding dirt road covered with fallen leaves and small stones, leading into the distance. On either side of the path are tall, slender trees with thin trunks. The ground is covered in dry, golden-brown grass and fallen leaves. The background is a soft, hazy blue, suggesting a misty or foggy day. The overall mood is serene and quiet.

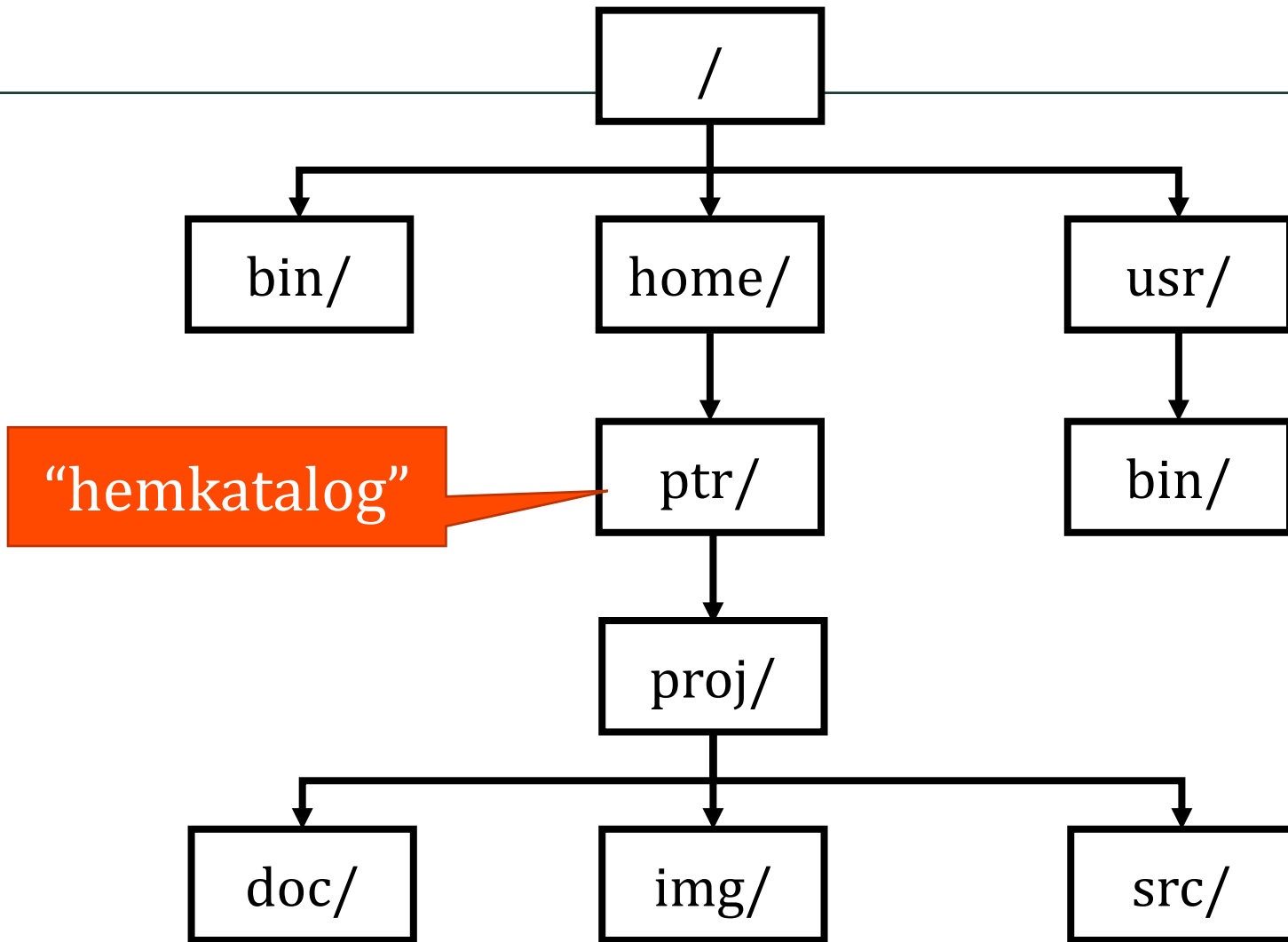
Sökvägar

Sökvägar i terminalen

- När vi har en terminal befinner vi oss någonstans i filsystemet
 - “(current) working directory” (CWD) - arbetskatalog
- `pwd` (print working directory) skriver ut CWD
- `cd` (change directory) byter CWD
- Fungerar både med absolut och relativ sökväg



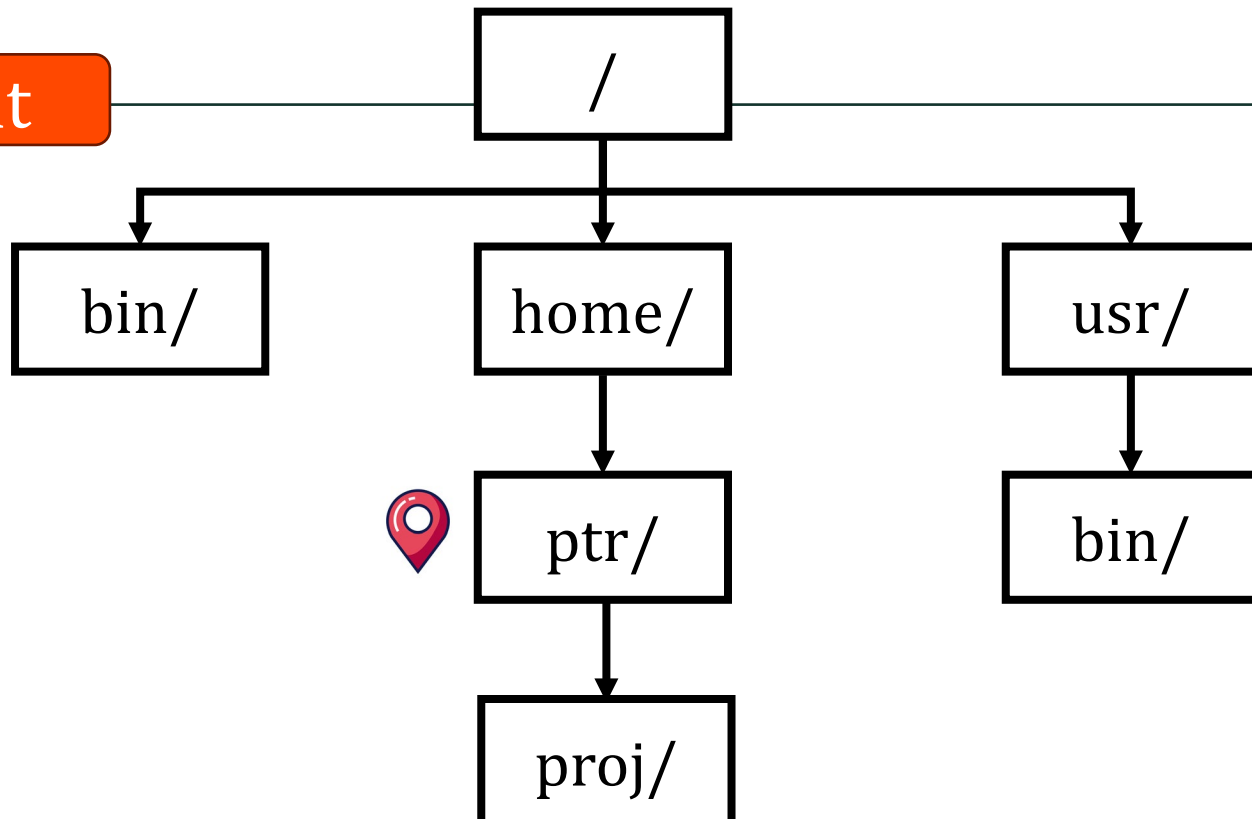




Sökvägar

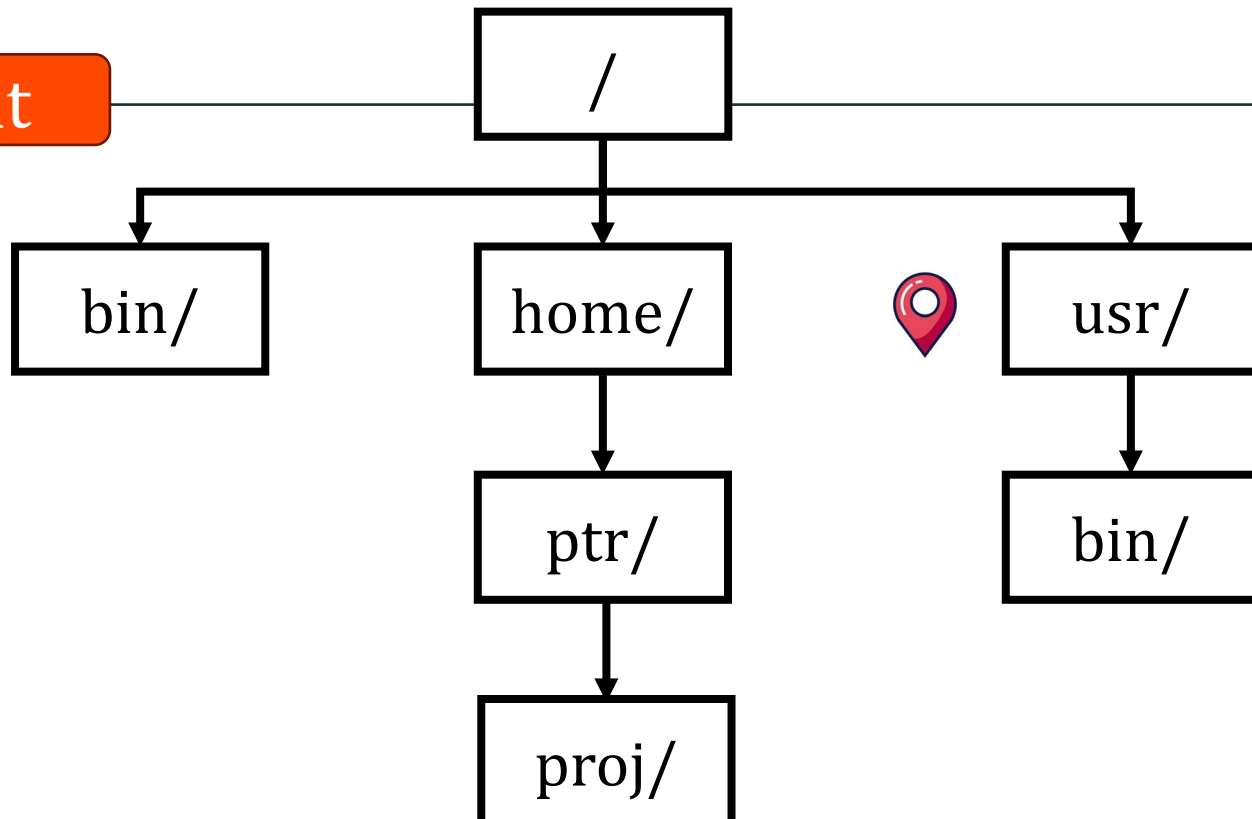
- Absolut sökväg
 - Hur tar man sig till X “från root”
 - Tänk GPS-koordinat
 - Börjar *alltid* med /
- Relativ sökväg
 - Hur tar man sig X “härifrån”
 - Tänk vägbeskrivning
 - Börjar *aldrig* med /

Absolut



/home/ptr

Absolut



/usr

Absolut



/

bin/

home/

usr/

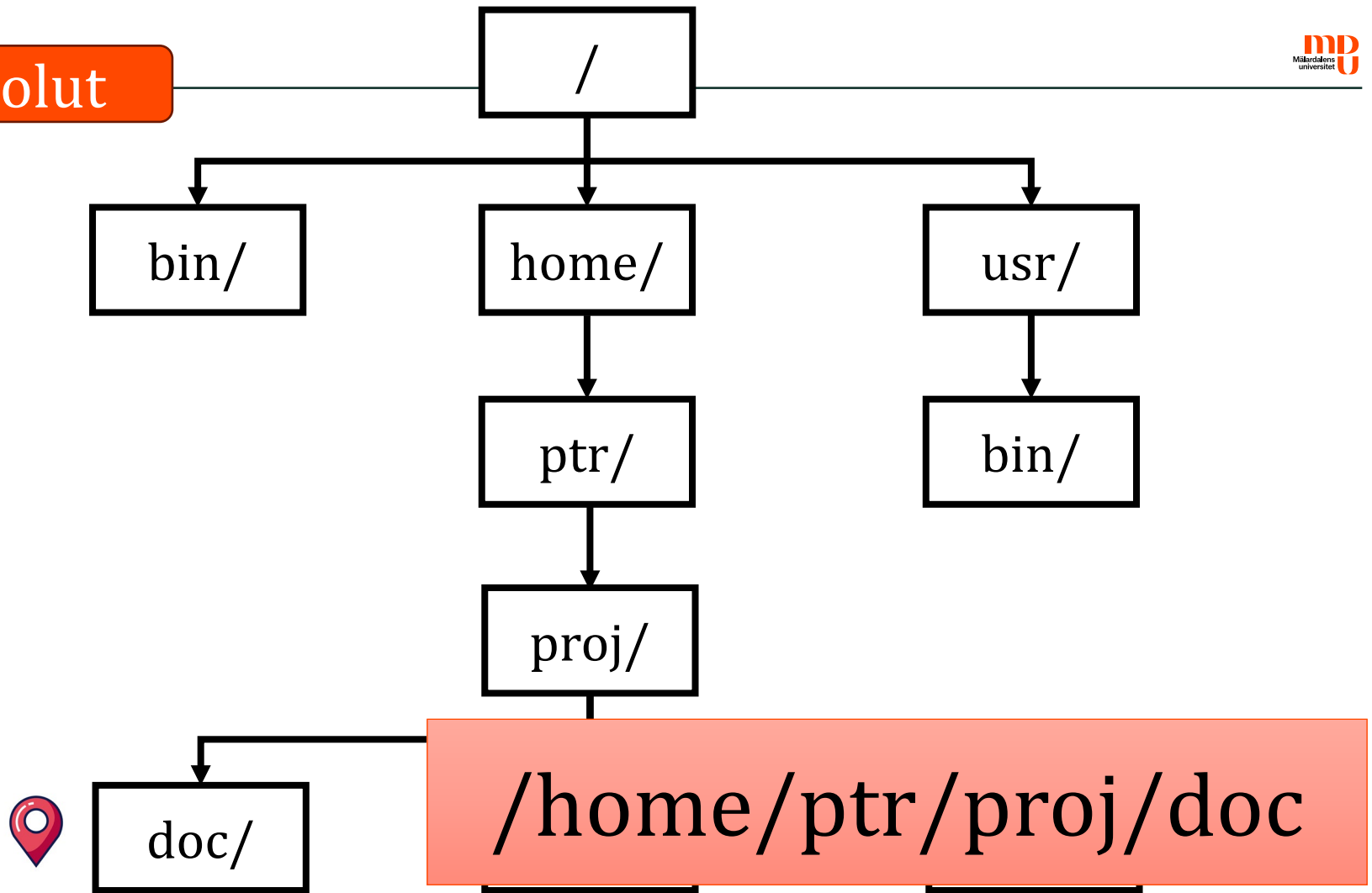
ptr/

bin/

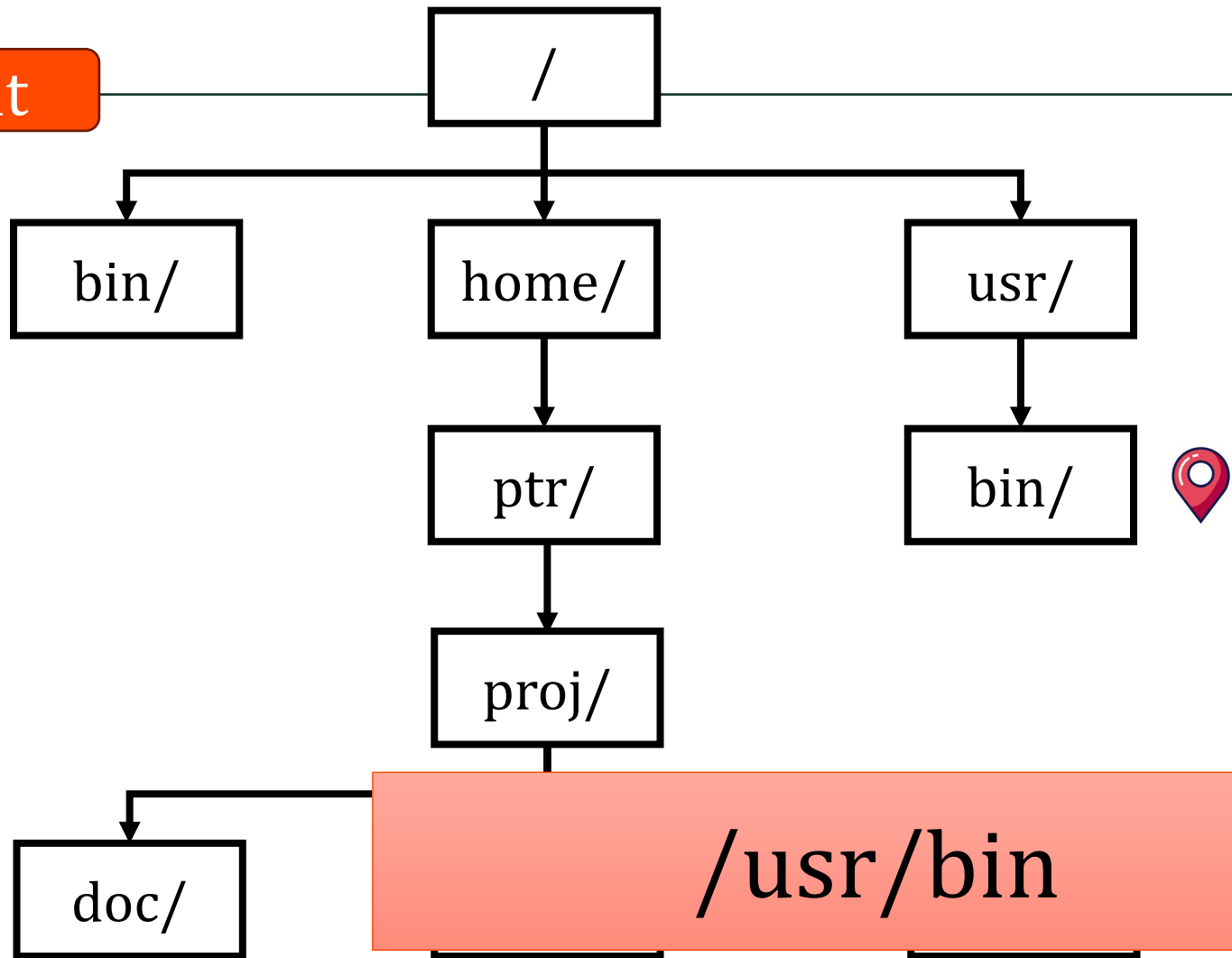
proj/

/

Absolut



Absolut



Absolut

/



bin/

home/

usr/

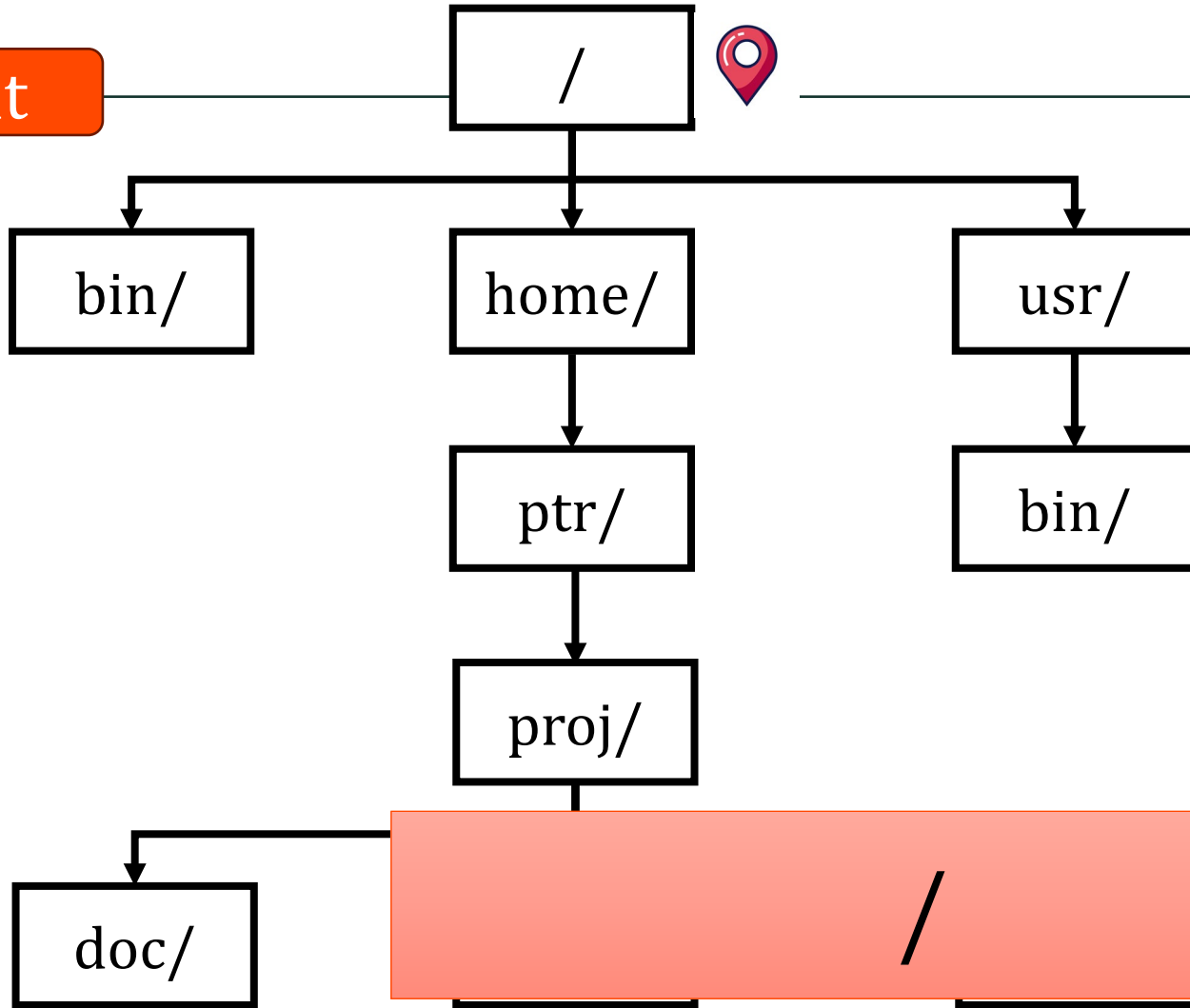
ptr/

bin/

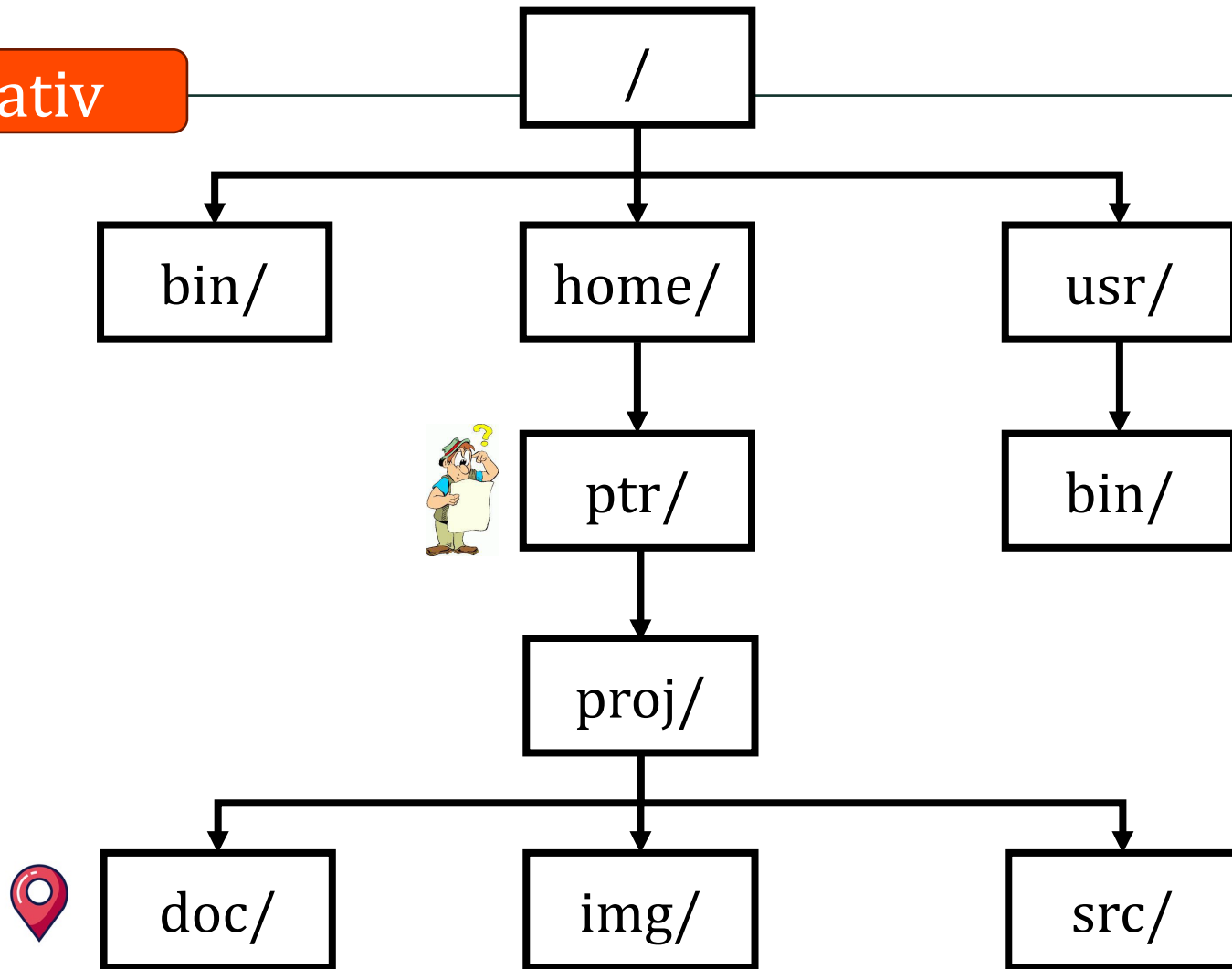
proj/

doc/

/



Relativ



Relativ

proj/doc

bin/

home/

usr/

ptr/

bin/

proj/

doc/

img/

src/



Relativ

proj/src

bin/

home/

usr/

ptr/

bin/

proj/

doc/

img/

src/



Relativ

ptr/proj/src

bin/



home/

usr/

ptr/

bin/

proj/

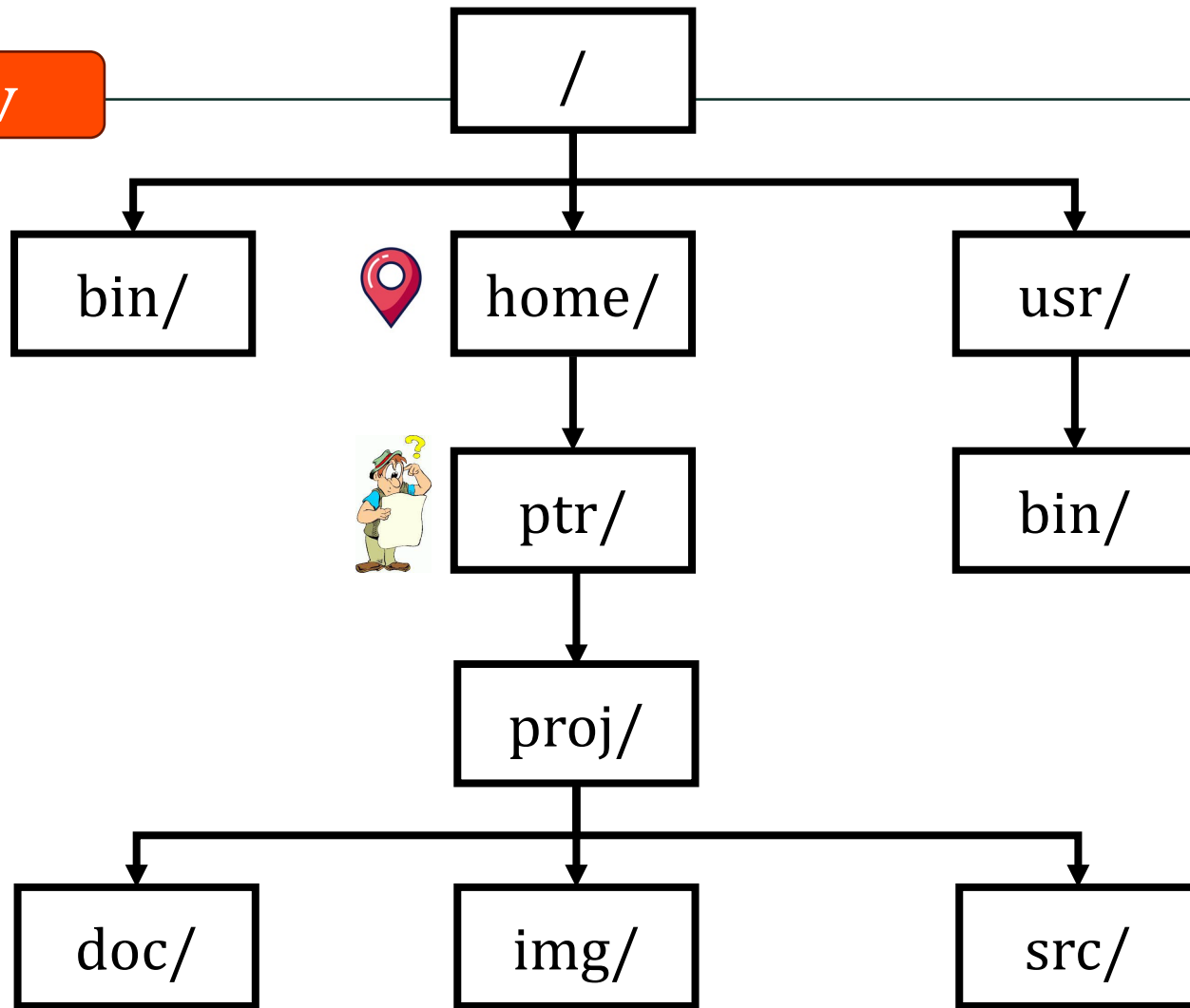
doc/

img/

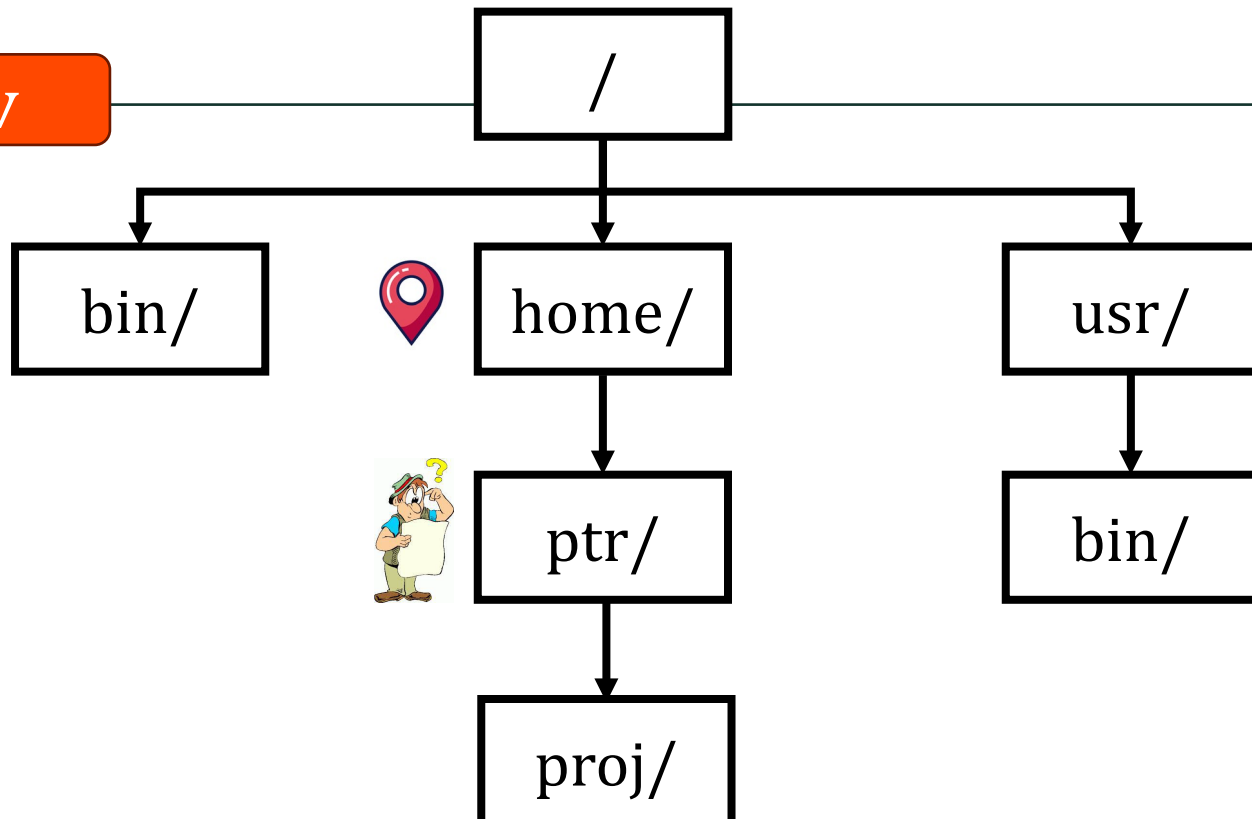


src/

Relativ



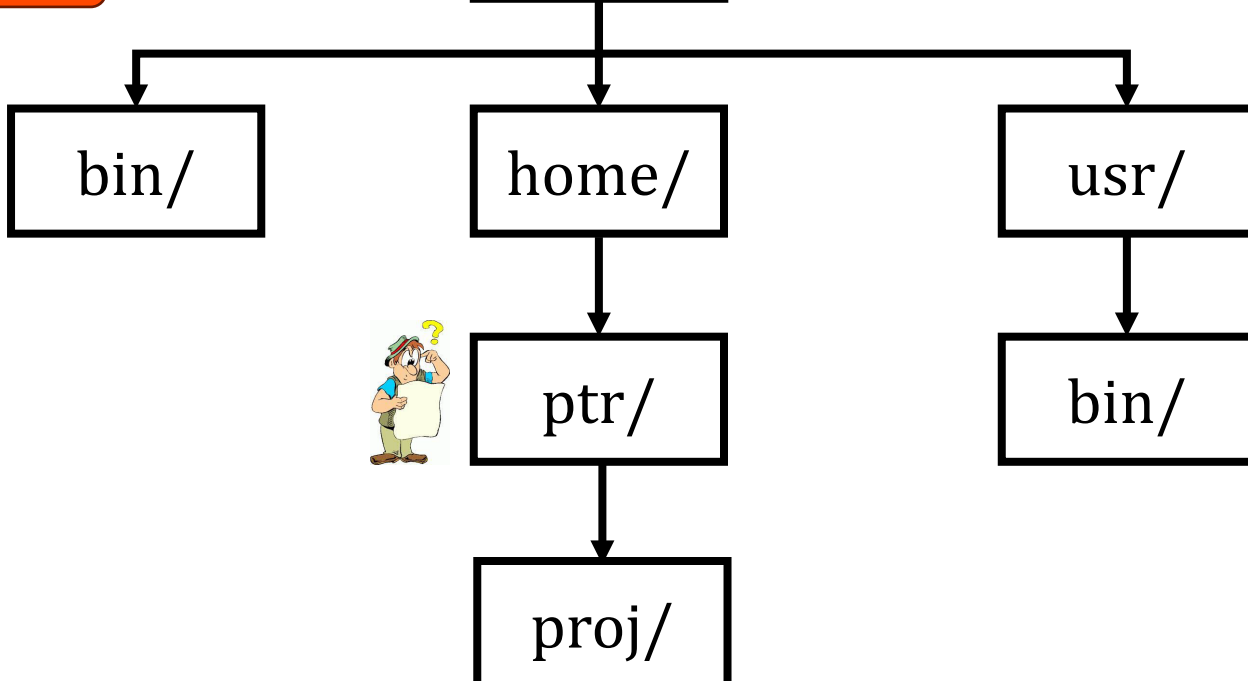
Relativ



Relativ

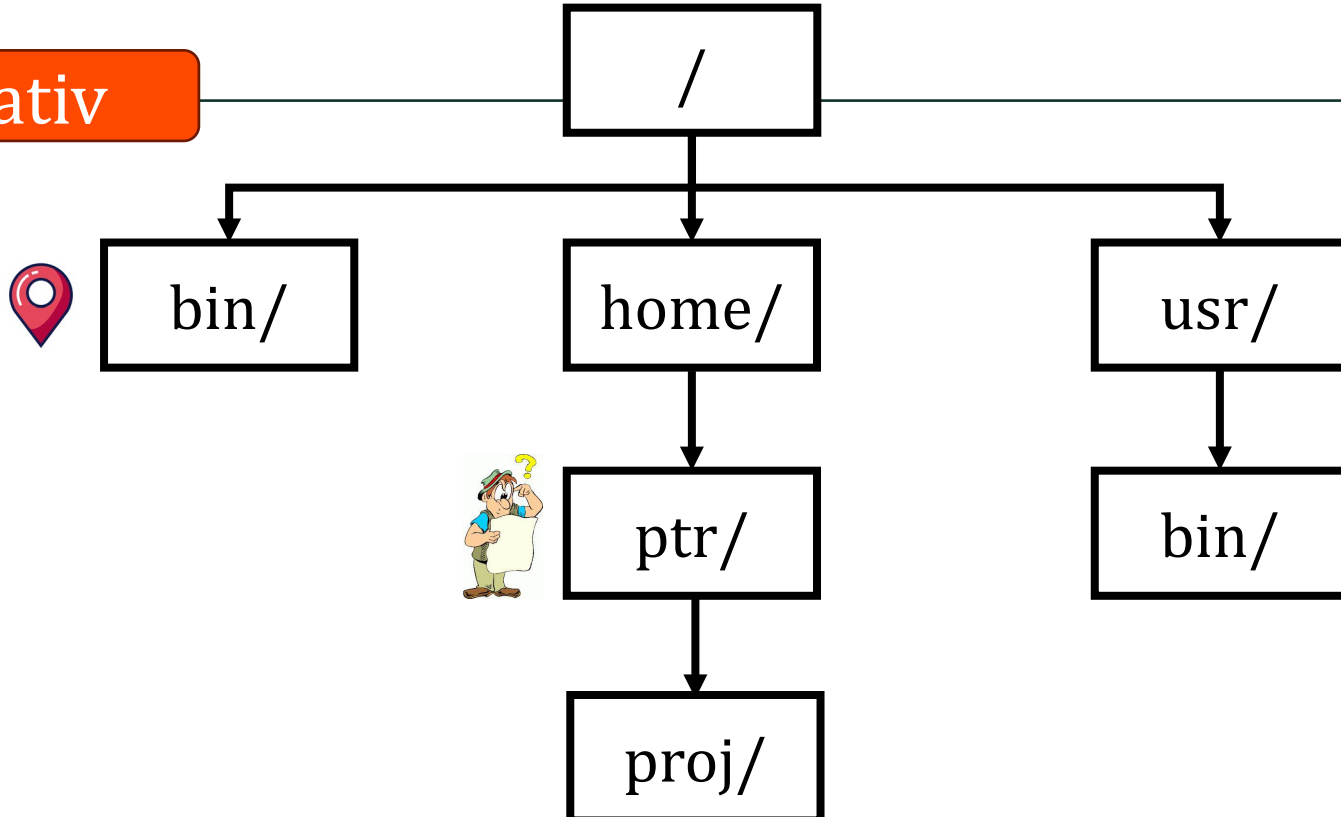


/



../..

Relativ



..../bin

Relativ

proj/doc

bin/

home/

usr/

ptr/

bin/

proj/

doc/

img/

src/



Relativ

../..

bin/

home/

usr/

ptr/

bin/

proj/

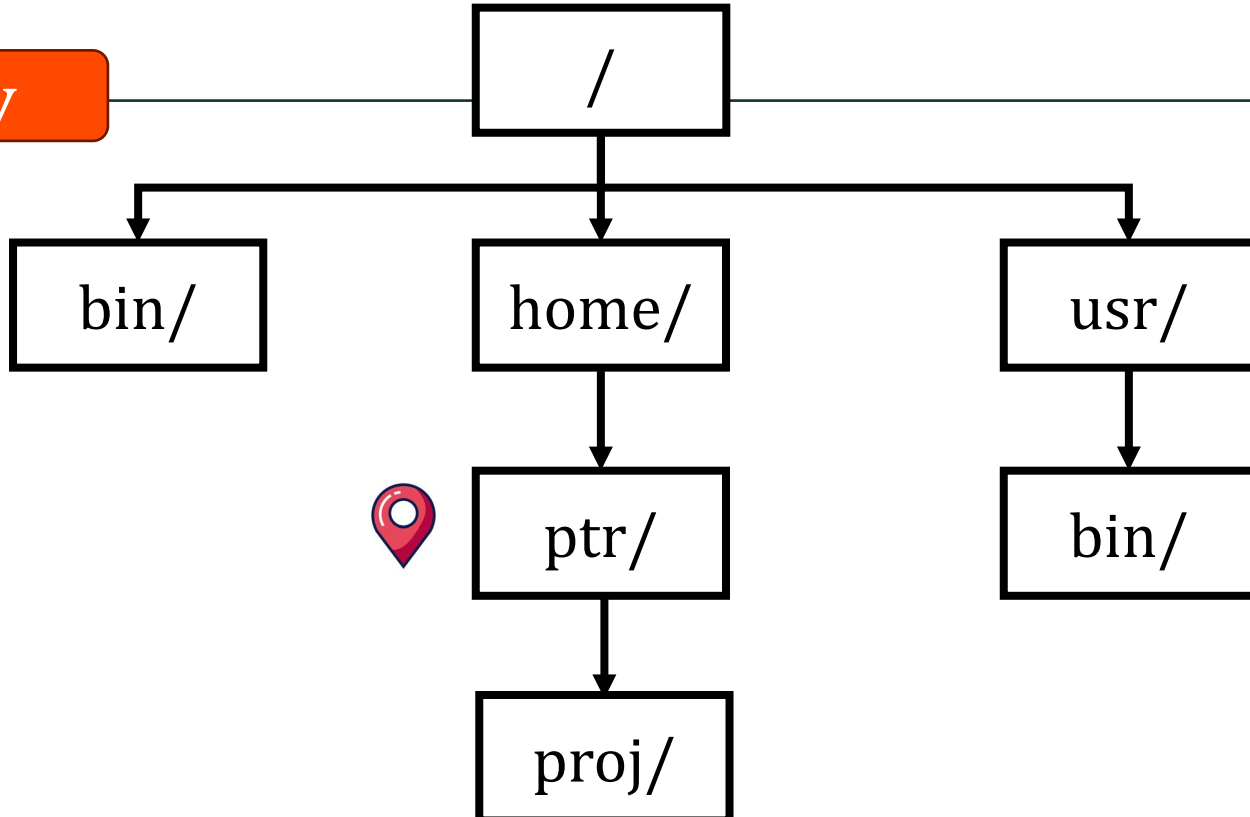
doc/

img/

src/



Relativ



../home/ptr

Flytta runt i systemet

- Om man har en lång sökväg kan man skriva hela på en gång
 - `cd home/ptr/proj/doc`
- Man kan även bryta upp och gå steg för steg:
 - `cd home/`
 - `cd ptr/`
 - `cd proj/`
 - `cd doc/`

Filhantering

Filhantering

- Skapa en fil: `touch`
 - Var hamnar filen?
 - Vad heter filen?
 - Vad innehåller filen?
- Egentligen för att ändra "senast ändrat"-tid.

Flytta & Kopiera & Ta bort

- Att flytta filer kallas “move”
 - mv
 - Används även för att döpa om (samma sak!)
- Att kopiera filer kallas “copy”
 - cp
- Att ta bort filer kallas “remove”
 - rm



(Globbing)

Ta bort flera saker på en gång

- Tänk dig att du har massor av kataloger:
 - `mapp1, mapp2, mapp3, mapp4`
- Vi vill ta bort allihopa på en gång
 - Kan skriva `rmdir mapp1, rmdir mapp2, ...`
 - Vill vi ta bort alla kataloger kan vi skriva `rmdir *`



Försiktigt! När man använder `*` är det väldigt lätt att ta bort för många filer!

Ta bort flera saker på en gång

- Tänk dig att du har massor av kataloger:
 - `viktig_mapp1, viktig_mapp2, viktigt_mapp3,`
 - `oviktig_mapp1, oviktig_mapp2, oviktig_mapp3,`
- Vi vill ta bort alla oviktiga mappar på en gång
 - Kan skriva `rmdir oviktig_mapp1, rmdir oviktig_mapp2, ...`
 - Vill vi ta bort alla oviktiga kataloger kan vi skriva `rmdir oviktig_*`

“Globbing”

- BASH expanderar `*` med allting som *matchar*
- En `*` kan bytas ut mot vilken text som helst
- Ensam `*` matchar allt!
- `oviktig_*` matchar allt som börjar med “`oviktig_`”

“Globbing”

- BASH expanderar `?` med exakt ett tecken som *matchar*.
- En `?` kan bytas ut mot vilket *tecken* som helst
- `oviktig_?` matchar allt som börjar med “`oviktig_`” och har exakt ett tecken till

Filsökning

Olika kommandon för att söka

- Leta på hårddisken nu
 - `find`
- Använder en databas över filer
 - `locate`
 - *updatedb* uppdaterar databas
- Även
 - `whereis`

Arkivering





Vad gör man med
massa mappar?



Vad gör man med
massa filer?





120 petabytes
= 120,000,000 GB

Pack ihop/upp filer till/från ett *arkiv*

- Kommandot "tape archive"
 - `tar`
- Packa ihop och packa upp filer
 - `tar cvf arkiv.tar fil1 fil2 fil3`
 - `tar xvf arkiv.tar`
- Create/eXtract, Verbose, File
 - Flaggor utan -

Komprimering



Om man har en jättestor fil...

- Kan man *komprimera*
- En fil komprimeras så den tar mindre plats
 - Kan bara användas efter den har extraherats
 - `gzip fil`
- Komprimerar filen `fil` till filen `fil.gz`

Lite om komprimering

- Komprimering fungerar (enkelt) så att samma data sparas på mindre plats, t.ex.:

row, row, row, your boat,
gently down the stream.
merrily, merrily, merrily, merrily,
life is but a dream.

- Kan istället beskrivas som

3*row, your boat,
gently down the stream.
4*merrily,
life is but a dream.

Kombinera arkivering och komprimering

- En flagga till `tar` för att automatiskt komprimera
 - `tar czf fil.tar.gz fil1 fil2 fil3`
- Arkiverar och komprimerar till filen `fil.gz`
- För att packa upp kan man använda flagga
 - `tar xzf arkiv.tar.gz`
 - fungerar också utan på nyare versioner

Skript

Vad är ett skript?

- En sekvens av kommandon
- Nu tittar vi på enkla skript (inga loopar, förgreningar, ...)
- Genom en texteditor skriver vi en kommand per rad
- För att köra: `bash skript.bash`

Skript vs. programmering

- Python
 - Båda körs rad för rad
- C
 - *C kompileras* till en körbar fil
- I denna kurs tittar vi på BASH-skript (d.v.s. för skalet BASH)
- För att köra: `bash skript.bash`

”she-bang”

- Vi börjar alla skript med en specifik rad:

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

- Inget magiskt, berättar bara var `bash` finns på datorn
- Så småningom lär man sig den utantill

Idag, massor av exempel

- Vi går igenom många exempel på skript
- Få en känsla hur man skriver och kör dem
- Senare labbar och speciell LAB2 består av mycket skriptning

Exempel: skriva ut hej

- Input: n/a
- Output: Skriver ut "Hej hej" i terminalen

Exempel: hello och goodbye

- Input: n/a
- Output: Skriver ut först "hello" och sedan "Goodbye" (på ny rad) i terminalen

Exempel: Hälsa på

- Input: ett namn
- Output: Skriver ut "Hello" och sedan namnet

Exempel: Lista filer i mapp

- Input: en mapp
- Output: Skriver ut en lista på alla filer i mappen

Exempel: Kopiera mapp

- Input: en mapp
- Output: n/a
- Side-effect: Kopierar mappen till en ny mapp med "_copy" på slutet av namnet

Exempel: Systeminformation

- Input: n/a
- Output: Systeminformation

Exempel: Klistra ihop

- Input: två filnamn: fil1 fil2
- Output: n/a
- Side-effect: skapar en ny fil som heter fil1_fil2 och innehåller först fil1 och sedan fil2

Exempel: Skapa mappar

- Input: n/a
- Output: n/a
- Side-effect: Skapar en mapp "project" som innehåller mappar "src", "img" och "doc"