Filer i Linux og Bourne-again shell

Filbegrepet

- En *fil** er en grunnleggende lagringsenhet i et OS
- Brukes for alle data som:
  - Lagres utenfor RAM (primærminnet)
  - På permanente media (sekundærminne)
- To hovedklasser – *datafiler* og *eksekverbare* filer
- Alle filer har et *navn* og en *type***

*: Engelsk “file”, kommer fra papirbaserte informasjonssystemer

**: Ofte angitt med en “extension” på slutten i filnavnet, f.eks. `.java`, `.txt`

To definisjoner av filer i Linux

- En samling data (bits og bytes) som *logisk* hører sammen
  - Tekstdokument, bilde, lydfil, video, programvare
  - Lagres typisk på en disk
- En kilde der data kan leses *fra*, eller et medium som data kan skrives *til*
  - Betyr at “alt er filer”, også f.eks. skjerm, mus og printer
  - Applikasjoner kan behandle I/O-enheter på samme måte som diskfiler

Filsystemet

- Den delen av et OS som kontrollerer hvordan data lagres på og hentes frem fra permanente media
- Data deles opp i individuelle deler, filer, som får hvert sitt navn/identifikator
- Filsystemet er *strukturen, reglene* og *mekanismene* som brukes for å håndtere filene
- Ulike OS og ulike lagringsmedia bruker ofte forskjellige typer av filsystemer

Oppgavene til filsystemet *

- Håndtere både filenes *innhold* og *metadata*
- Strukturere lagringsplassen som er tilgjengelig
- Gi god pålitelighet, effektivitet og datasikkerhet
- Tilby brukerne verktøy for å håndtere filer
- Muliggjøre deling av data
- Backup og redundans (?)

*: Vi skal lære mer om filsystemer senere i kurset

Filer og hardware

- Filer lagres på *permanente* media (også kalt sekundærminne), typisk harddisker
- Lagrer store datamengder til lav kostnad
- Ikke direkte tilgjengelig for CPU
- Aksesseres gjennom egne I/O-celler
- Data overføres oftest til/fra RAM av dedikert hardware* i store *blokker* for å øke hastigheten

* DMA-brikker (Direct Memory Access) som flytter data direkte uten å gå om CPU

Noen typer permanente media

- Sortert etter lese-/skrivehastighet:
  - Solid-state drive (SSD/flash-disk)
  - Magnetiske, roterende disker (hard drives)
  - Optiske disker
  - Magnetisk tape

Det fysiske filsystemet

- Filsystemet er fysisk bygget opp av maskinvare, elektronikk og lavnivå programvare:
  - Disker og andre lagringsenheter
  - Elektronikk og elektromekanikk (lese- og skrivehoder, rotasjonsmotorer…)
  - Blokkbaserte I/O-enheter og disk-kontrollere
  - Magnetiske spor og sektorer på overflaten av disker
  - Tabeller med “fil-pekere” som lagrer fysiske adresser for start og slutt på (deler av) filer

OS og det logiske filsystemet

- Et OS gir brukerne abstraksjon *vekk* fra hardware og det fysiske filsystemet
- Tilbyr et sett med logiske brukerkommandoer for filhåndtering
- Oversetter kommandoer til fysisk håndtering av media internt i OS
- Brukere av maskinen trenger ikke kjenne til detaljer om hardware og det fysiske filsystemet

Litt terminologi for filsystemer *

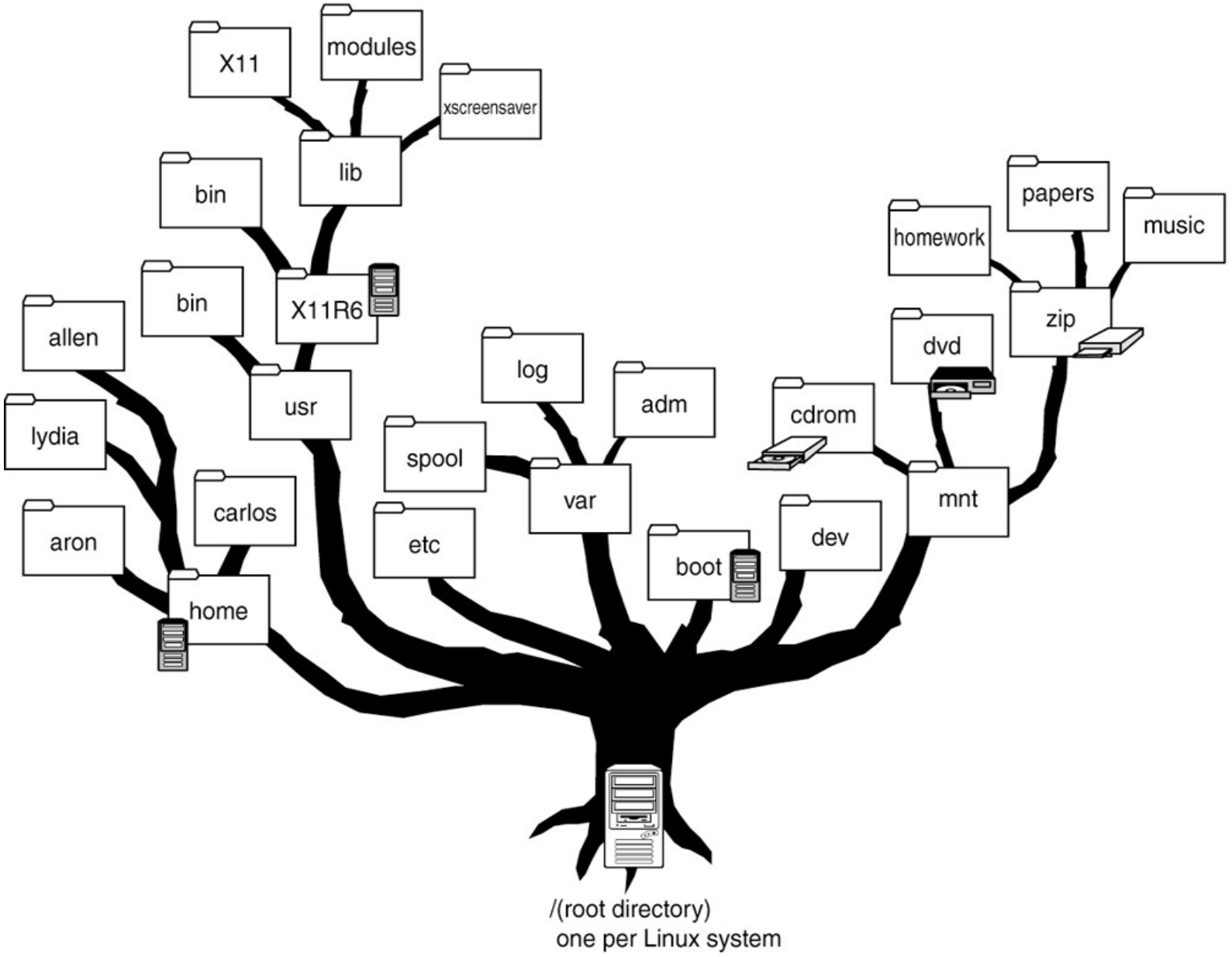
- Globalt filsystem:
  - Samlingen av alle kataloger på alle lagringsenheter som er tilknyttet maskinen
- Partisjon:
  - Logisk oppdeling av fysisk lagringsenhet i uavhengige enheter
- Katalog (directory/mappe):
  - Bruker-spesifisert logisk samling av flere filer
  - Gir et hierarkisk organisert filsystem (“mapper inne i mapper”)
- Link:
  - Peker fra en katalog eller fil til en annen, for enklere filtilgang

*: Terminologien her er ikke helt presis, vil bli forklart nærmere senere i kurset

Det *globale* filsystemet i Linux

- Alle Unix-lignende OS har et logisk / virtuelt filsystem som er *globalt*
- Alle filer på hele systemet ser ut til å ligge i *ett enkelt* hierarkisk katalogtre *
- Det er bare ett toppnivå (rotkatalogen) som kalles *root* og betegnes med '/' (tegnet skråstrek, “slash”)

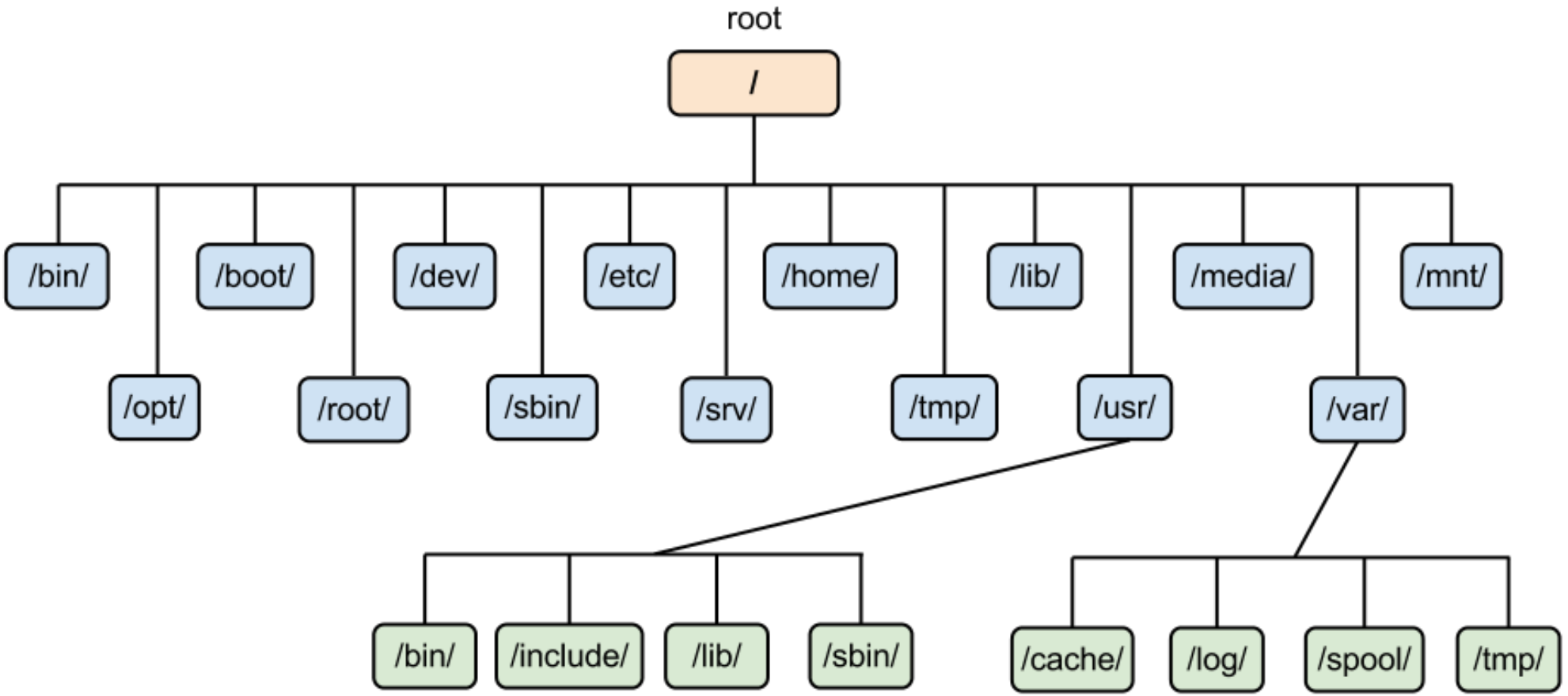
*: I motsetning til f.eks. Windows som gjerne har flere katalogtrær navngitt med bokstaver



Det globale filsystemet kan være satt sammen av flere mindre filsystemer (partisjoner), på samme eller på ulike device

Standard oppbygging av Linux filsystem

Se avsnitt 3.6 i læreboken (eller f.eks. howtogeek.com) for en gjennomgang av innholdet i toppnivåkatalogene i Linux



Søkesti: Angir hvor en fil ligger

- En søkesti er en liste av katalognavn skilt med tegnet `/` (skråstrek)
- Absolutt søkesti:
  - Starter på *toppen av filsystemet*, i toppnivåkatalogen `root` , som angis med `/`
  - Går helt frem til katalogen der filen vi skal bruke befinner seg
- Relativ søkesti:
  - Stien starter i *stående katalog* og går frem til katalogen der filen befinner seg

Søkestier og Linux-kommandoer *

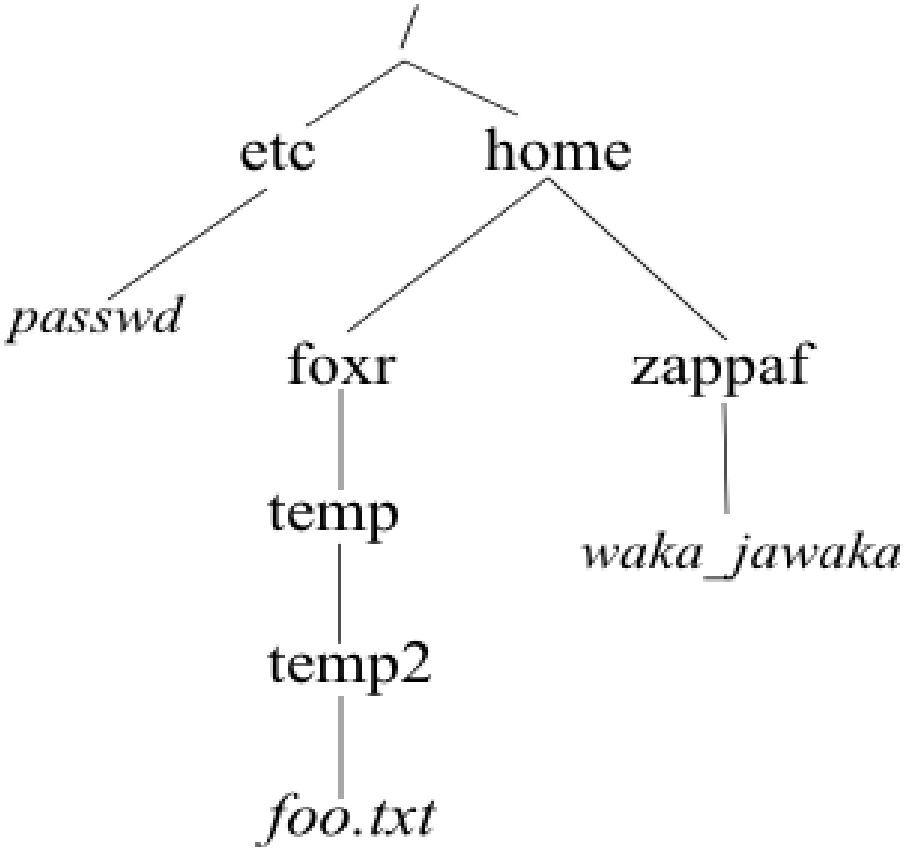
- Input til Linux-kommando er typisk en fil eller katalog
- Bashinterpreteren må vite hvor den finner:
  - Den eksekverbare filen som svarer til kommandoen som skal utføres (f.eks. `/bin/ls`)
  - Filene som brukes som input og output
- Filene kan ligge i en hvilken som helst katalog
- Vi må enten angi *søkestien* (search path) til filene, eller legge inn kataloger i miljøvariabelen `PATH`

*: Filer kan også håndteres med pek-og-klikk / drag-and-drop i et GUI for Linux, omtrent på samme måte som i Windows og OS X

Spesialtegn i søkestier

`.`	Stående katalog
`..`	Katalogen ett nivå over stående katalog
`~`	Innlogget brukers hjemmekatalog (typisk `/home/brukernavn`)
`~bruker`	En annen brukers hjemmekatalog

Eksempel fra læreboken



Antar at stående katalog er `temp2`:

- Absolutt sti til katalogen `zappaf`:
  `/home/zappaf`
- Relativ sti til katalogen `zappaf`:
  `../../../zappaf`
- Absolutt sti til filen `passwd`:
  `/etc/passwd`
- Relativ sti til filen `passwd`:
  `../../../../etc/passwd`
- For å utføre programmet `waka_jawaka`:
  `/home/zappaf/waka_jawaka`
  `../../../zappaf/waka_jawaka`
- Hvis stående katalog er `/home/zappaf`:
  `./waka_jawaka`

Ta ut deler av søkestier

- Linux tilbyr kommandoer som tar en søkesti som input og returnerer *deler* av den
- Nyttig i bl.a. shellprogrammering
- `basename` *`søkesti`*
  - Returnerer bare navnet på selve filen (eg. siste element i søkestien)
- `dirname` *`søkesti`*
  - Returnerer søkestien *frem til* der hvor filen ligger

Linux filtyper *

1. Regulære filer
   - Vanlige bruker-/systemfiler på disk
2. Katalogfiler
   - Inneholder en liste av filene (og underkatalogene) som ligger i en katalog – muliggjør hierarkisk filsystem
3. Spesialfiler
   - Devicefiler (block og character)
   - Pipes og sockets (interprosesskommunikasjon)
   - Links (hard og soft)

* : Kommandoen `file` kan brukes for å få informasjon om bl.a. filtype

Fil- og katalognavn i Linux

- Filsystem-kommandoene bruker *navn* til å identifisere filer og kataloger
- Filnavn kan inneholde alle tegn unntatt `/` og NULL-tegnet
- Lengde på filnavn vanligvis opptil 255 tegn
- Ingen regler for “extensions” eller formatering

Vanlig navnekonvensjoner for filnavn i Linux

- Unngå whitespace
- Unngå spesialtegn for shell som `$ * @ ! < > |`
- Unngå særtegn som f.eks. norske `æ ø å`
- Bruk: `0-9 a-z A-Z . - _`
- Bruk i størst mulig grad små bokstaver

File name generation (globbing)

- Kan angi *samlinger* med filnavn med spesialtegn (jokernotasjon, wildcards) i shellet:

  `*` Matcher en vilkårlig tegnsekvens

  `?` Matcher et vilkårlig tegn

  `[ ]` Et av tegnene mellom firkantparantesene

  `[ ! ]` Et av tegnene *ikke* mellom firkantparantesene

- Shellet leser spesialtegnene og utvider dem til en liste med filnavn, før kommandoen som skal håndtere filene utføres

File name generation: Eksempler

`ls *`	List alle filer i stående katalog
`ls a*`	Alle filnavn som begynner med a
`ls a*.txt`	Alle filnavn som begynner med a og slutter med .txt
`ls a?.txt`	`aa.txt ab.txt ac.txt a4.txt` …
`ls a[bc4].txt`	`ab.txt ac.txt a4.txt`
`ls a[0-9]*`	Alle filnavn som begynner med a etterfulgt av ett siffer
`ls [!a]*`	Alle filnavn som ikke begynner med a

Brace expansion

- Genererer alle mulige kombinasjoner av en tekst-streng og en liste med ord/items (ligner for-løkker)
- Eksempel:

  ```
  streng{ord1,ord2,ord3}
  ```

  ekspanderer til:

  ```
  strengord1 strengord2 strengord3
  ```

- Kan typisk brukes til å spare skriving og gå gjennom lister av filer og kataloger på en systematisk måte

Brace-expansion: Eksempler

- Gå gjennom flere kataloger med én kommando:

```
ls /home/bash/test/{foo,bar,baz,cat,dog}
```

- “Massenedlasting” av filer fra nettet:

```
wget http://domain.com/book/page{1,2,3,4,5,6}.html
wget http://domain.com/book/page{1..6}.html
```

- Skrive ut alle mulige kombinasjoner av en stor bokstav og et siffer (nesting av braces):

```
echo {A..Z}{0..9}
```