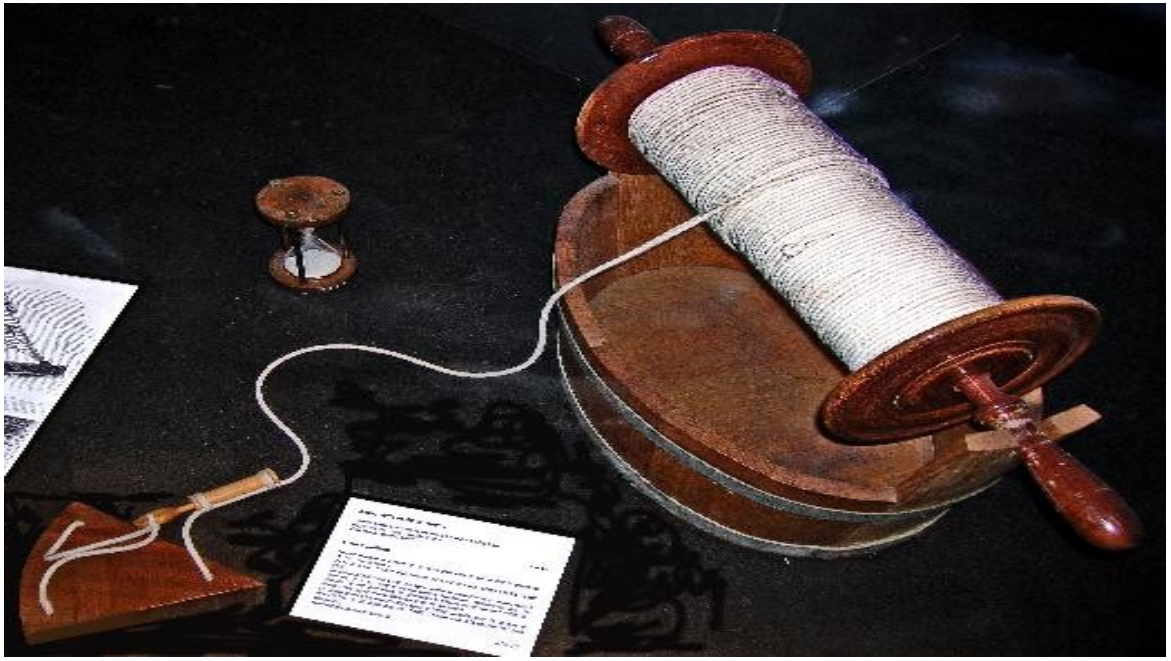
History



Definition

Data logging is the practice of recording sequential data, often chronologically. (Wikipedia)

Recorded information should be:

- Simple
- Traceable
- Maintainable

Logging Frameworks for Java

- java.util.logging (JUL)
- log4j
- logback
- Jakarta Commons Logging (JCL)
- Simple Logging Facade for Java (SLF4J)

Java Logging API

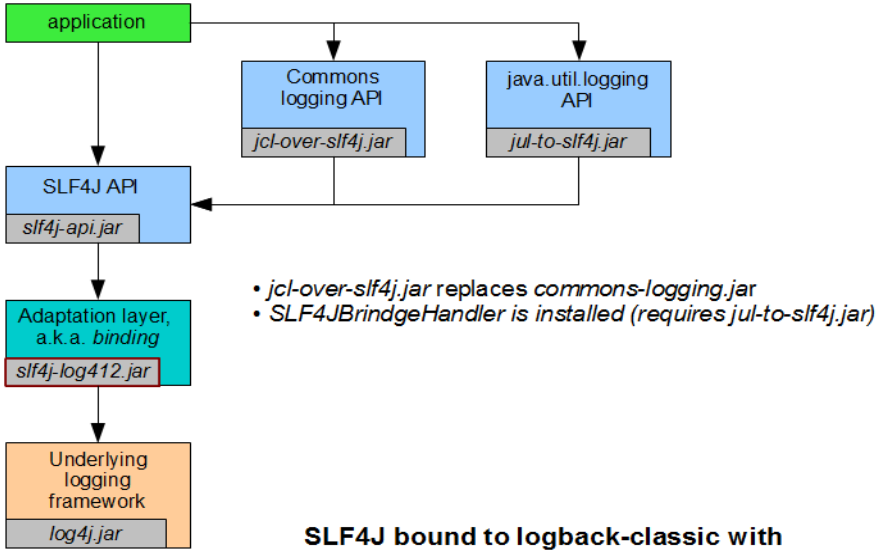
- Logger
- LogRecord
- Handler
- Level
- Filter
- Formatter

Level

- FATAL
- ERROR
- WARNING
- INFO
- DEBUG
- TRACE

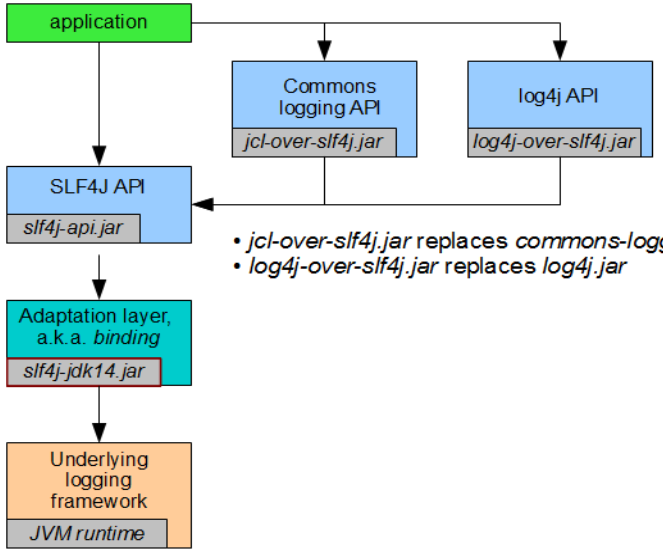
All messages of a level or higher are logged, e.g.: level WARNING also contains ERROR and FATAL

SLF4J bound to log4j with redirection of commons-logging and jul calls to SLF4J



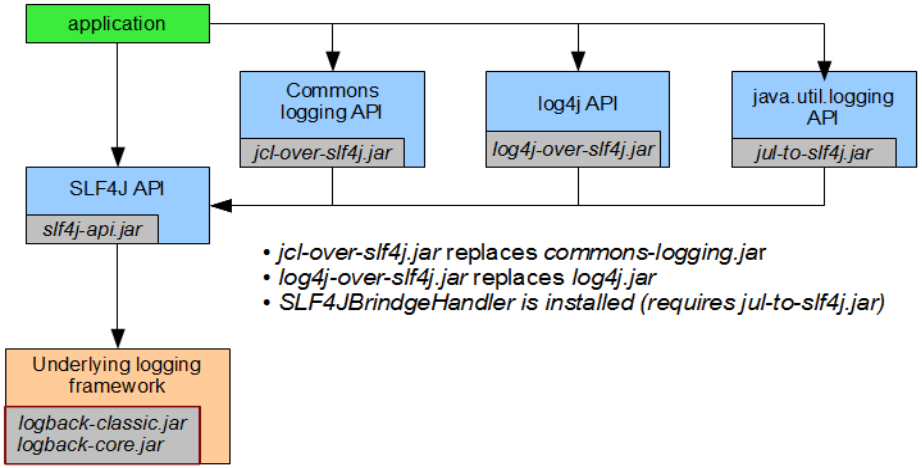
- *jcl-over-slf4j.jar* replaces *commons-logging.jar*
- *SLF4JBrindgeHandler is installed (requires jul-to-slf4j.jar)*

SLF4J bound to java.util.logging with redirection of commons-logging and log4j calls to SLF4J



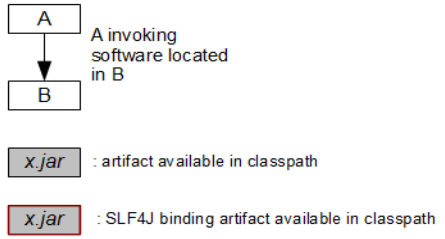
- *jcl-over-slf4j.jar* replaces *commons-logging.jar*
- *log4j-over-slf4j.jar* replaces *log4j.jar*

SLF4J bound to logback-classic with redirection of commons-logging, log4j and jul calls to SLF4J



- *jcl-over-slf4j.jar* replaces *commons-logging.jar*
- *log4j-over-slf4j.jar* replaces *log4j.jar*
- *SLF4JBrindgeHandler is installed (requires jul-to-slf4j.jar)*

These diagrams illustrate *all* possible redirections for various bindings for reasons of convenience and expediency. Redirections should be performed only when necessary. For instance, it makes no sense to redirect java.util.logging to SLF4J if java.util.logging is not being used in your application.



Summary

- logging records sequential data chronologically
- various logging frameworks exist
- different log levels are offered
- uses logger, handler, formatter

Contents

File Manipulation

```
import java.io.*;

public class Launcher {

    public static void main(String[] args) {

        File d = new File("blu/bla/testdir");
        // Create directory and all of its non-existent parent directories.
        d.mkdirs();

        if (d.isDirectory()) {
            System.out.println("The created directory is valid.");
            System.out.println("Its name is: " + d.getAbsolutePath());
            try {
                File f = new File("blu/bla/testdir/test.txt");
                f.createNewFile();
                File n = new File("blu/bla/testdir/newname.txt");
                f.renameTo(n);
            } catch (Exception e) {
                System.out.println("Error: The file creation failed.");
            }
        } else {
            System.out.println("Error: The directory creation failed.");
        }
    }
}
```

File

Punched Card Deck (File)

Electronic File

File Type

executable (programme)

- machine language
- script language
- bytecode

non-executable

- source code
- text
- image
- audio
- video
- database

directory

- container

pseudo

- device (e.g. /dev/printer, /dev/mouse)
- process information (e.g. /proc/68/environ)

Attribute / Property / Meta Data

- type
- size
- rights: owner, group, rwx, Access Control List (ACL)
- date of creation / last modification / last access
- hidden, system, archive
- comment, label
- record length/ format/ block size
- expiration date

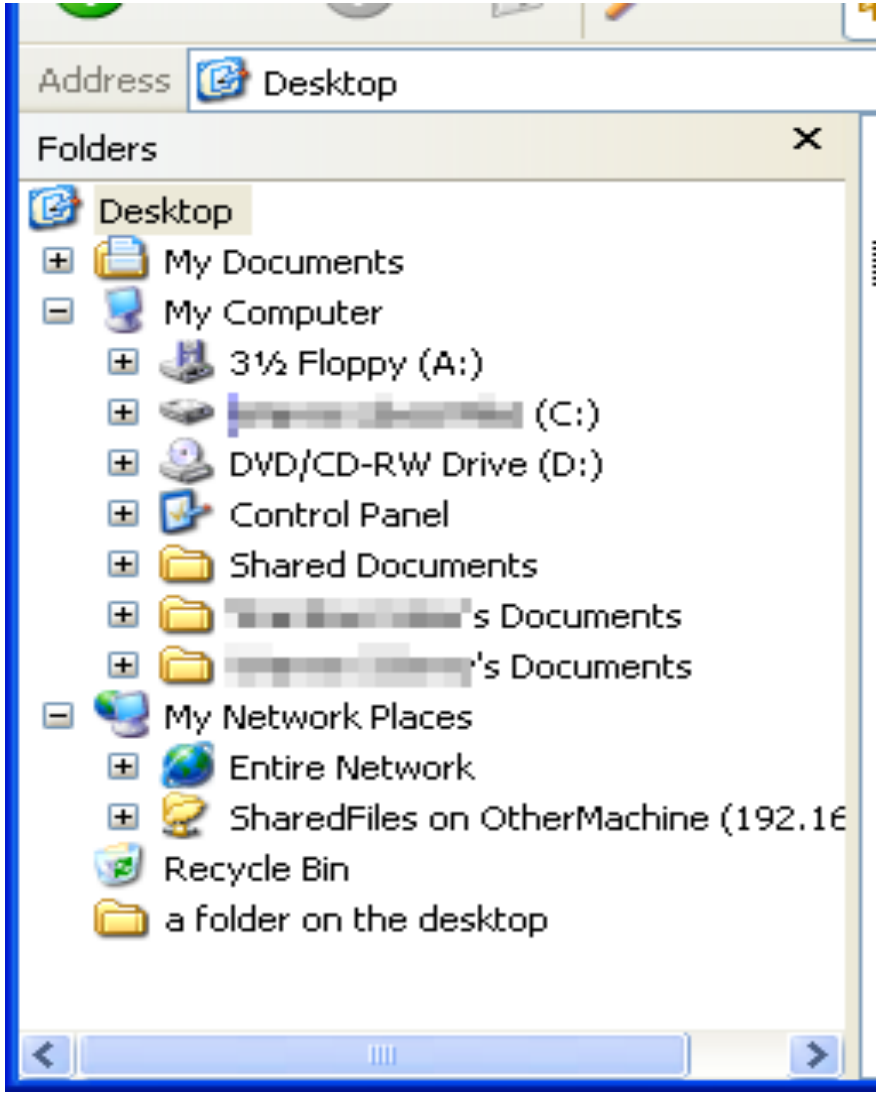
Naming

- well-formedness: depends on type of computer system
- historically: only a few letters or digits permitted
- meanwhile: long names (up to 255 characters) allowed
- elements: almost any unicode letter or digit
- spaces: allowed by some operating systems
- case-sensitivity: determined by the file system

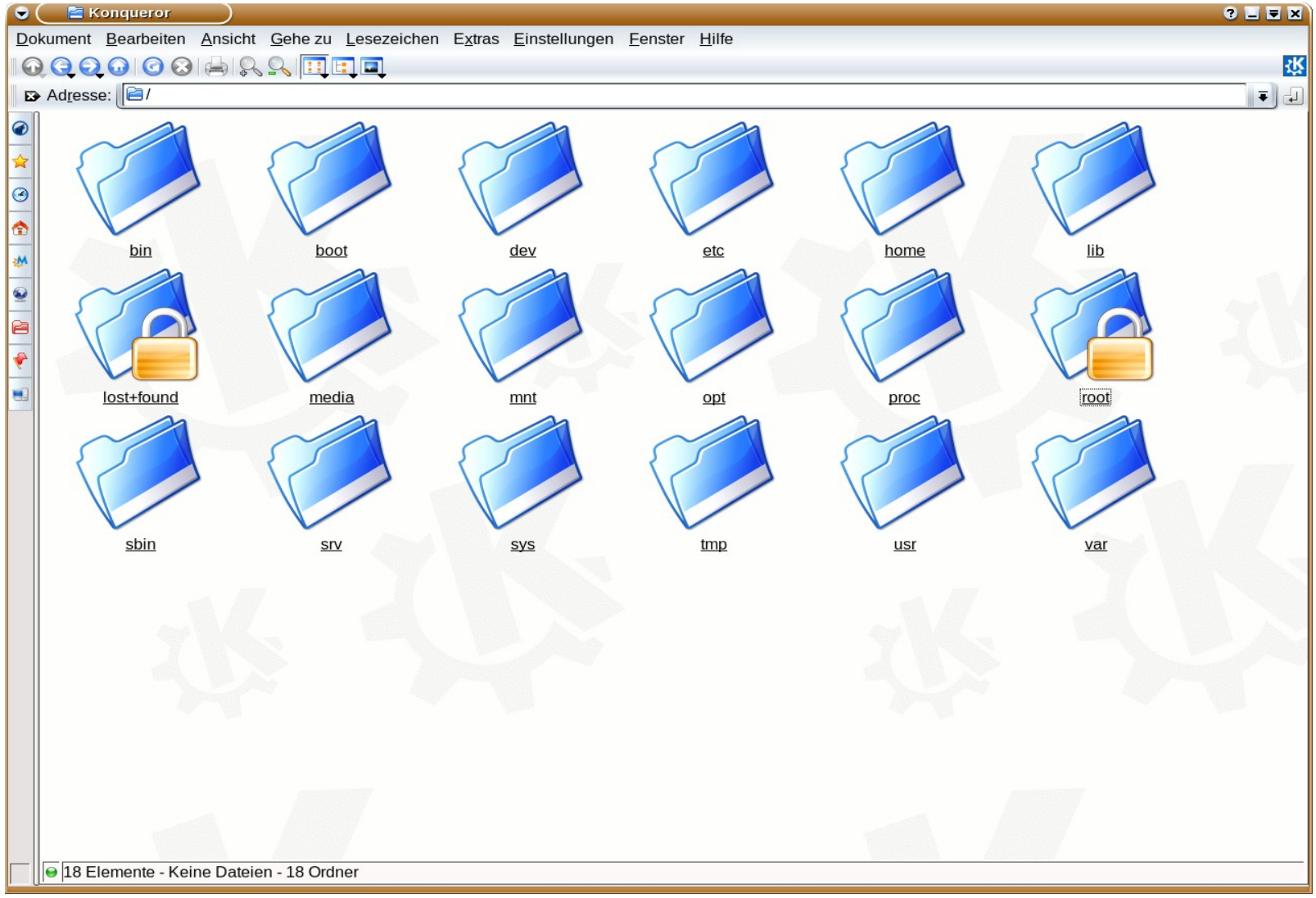
namensrichtlinie.pdf

File System

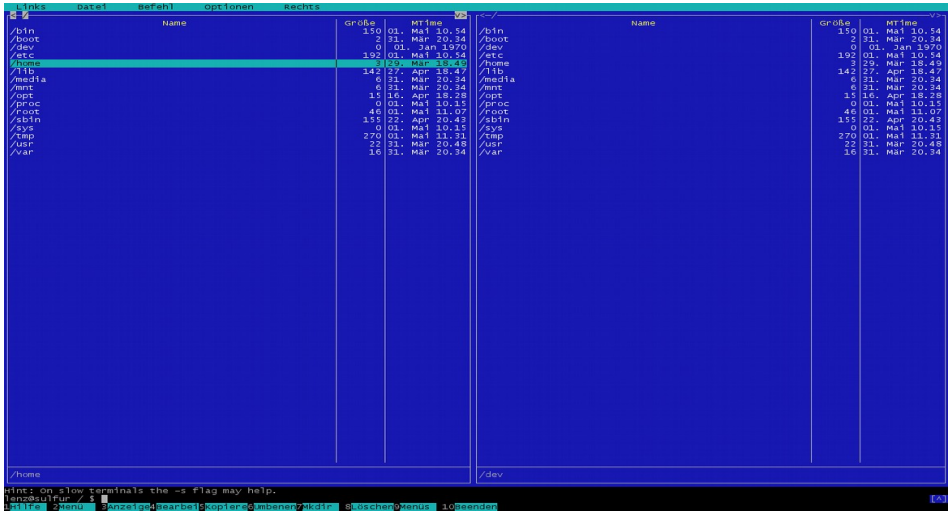
- root directory: /, C:, D:, E:
- directory: /bin/, /dev/, /etc/, /lib/, /usr/, C:\My Documents\
- sub directory: /usr/local/, /usr/src/linux/, C:\My Documents\My Pictures\
- file: usr/local/readme.txt, dev/hda1, C:\My Documents\My Pictures\VacationPhoto.jpeg

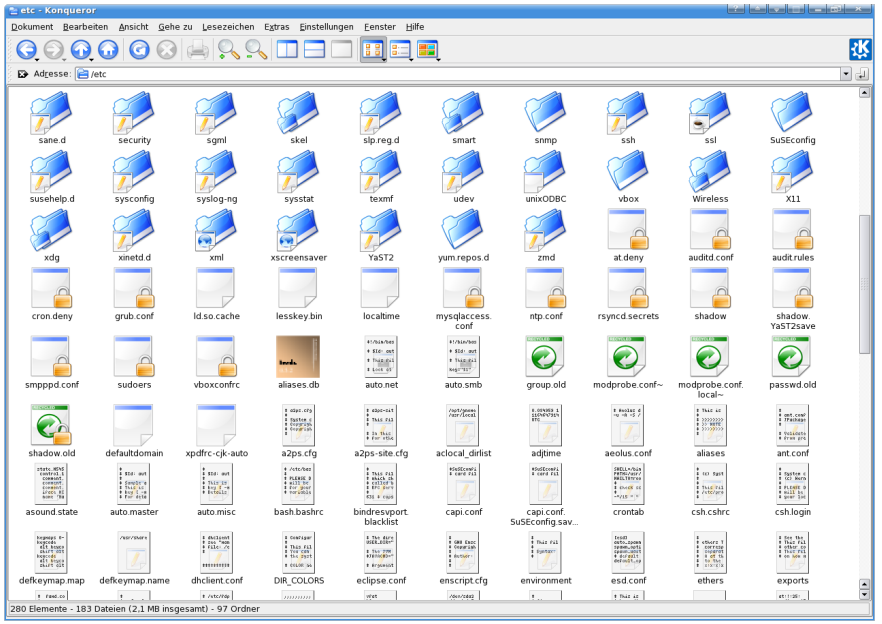


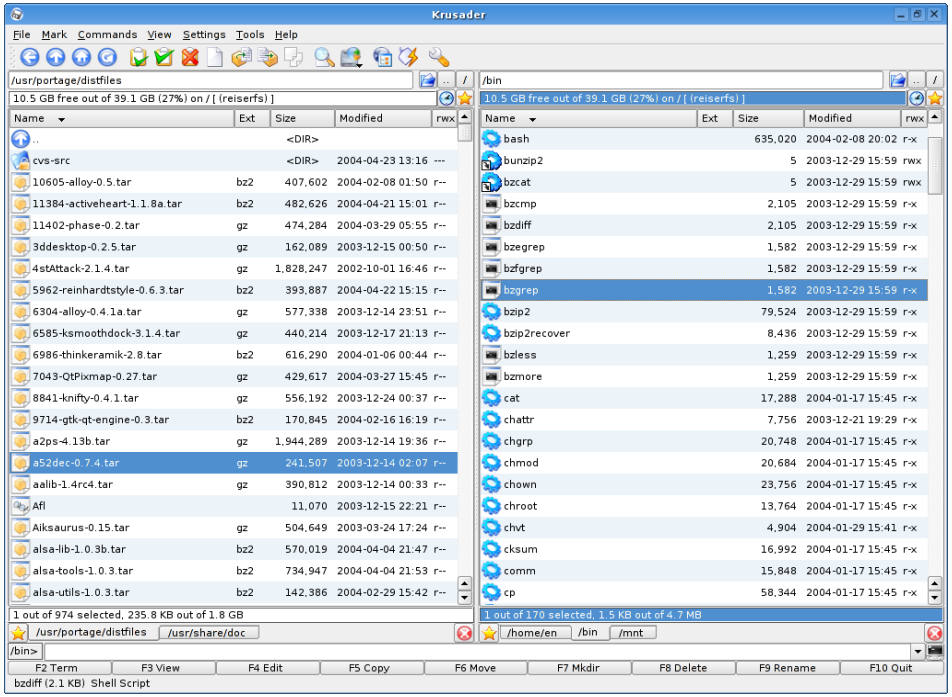
Filesystem Hierarchy Standard (FHS)

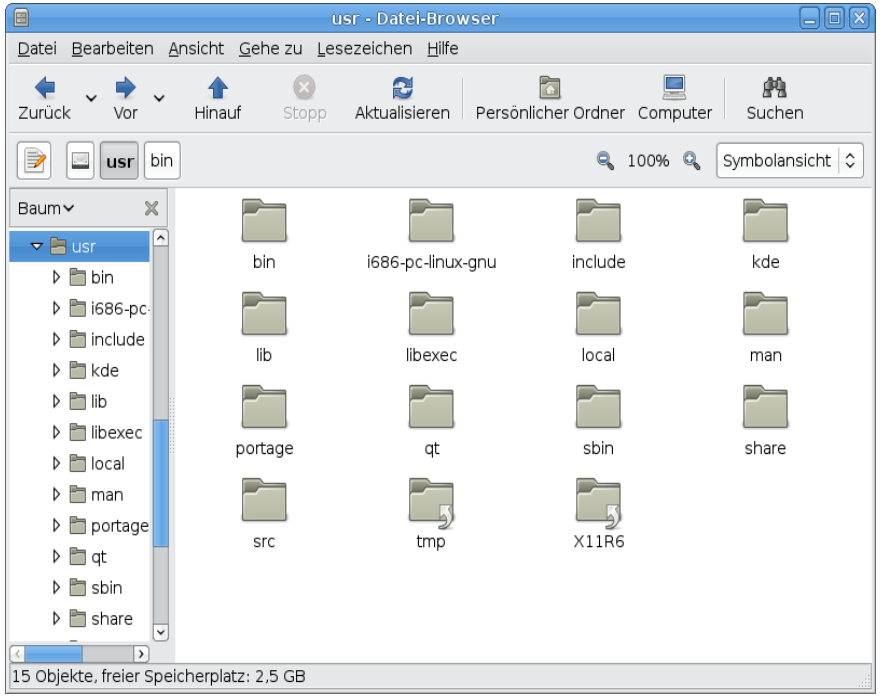


File Manager









Summary

- file represents a block of arbitrary information
- its attributes contain meta data
- directory is a special file that may contain other files
- files are organised in a file system (by operating system)
- file managers ease working with files

Contents

Example

```
import java.util.*;

public class Launcher {

    public static void main(String[] args) {

        List<Object> l = new ArrayList<Object>();
        // List<String> l = new ArrayList<String>();
        // List l = new ArrayList();
        // ArrayList l = new ArrayList();

        l.add("This is a String");
        l.add(new Short((short) 12));
        l.add(new Integer(35));

        for (Iterator<Object> i = l.iterator(); i.hasNext(); ) {

            System.out.println(i.next());
        }

        // for (Object o : l) {
        //
        //     System.out.println(o);
        // }
    }
}
```