

Introduction to Mathematical Software 2009



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Organisation

- Involved People
 - PD Dr. rer. nat. Ulf Lorenz
Email: lorenz@mathematik.tu-darmstadt.de
Room S2 | 15 230
Office hours: Tuesday, 9:30 – 10:30
 - Dipl.-Math. Christian Brandenburg
Email: brandenburg@mathematik.tu-darmstadt.de
Room S2 | 15 229
Office hours:
 - Thorsten Ederer
Email: ederer@mathematik.tu-darmstadt.de
 - Technical support:
 - <http://www.rbg.informatik.tu-darmstadt.de> (C 119)
 - Sebastian Hamann, probably in S2 | 15 317
times to be announced



Organisation

- **Topics**
 - We develop mathematical software
 - UNIX/Linux
 - C / C++ programming
 - We use mathematical software
 - Mathematics with a computer, Maple

- **Schedule**
 - Lectures: 14.10., 21.10., 11.11., 9.12., X-mas, ...
 - Exercises: 26.10., 2.11., 16.11., 23.11., 30.11., 14.12, X-mas, ...

- **Rooms**
 - Wednesday, 16:15 – 17:55 --- S103/123
 - Monday, 8:00 – 9:40 --- S2|02 C005 (Piloty building)

Organisation



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Additional Information
 - Webpage:
<https://www3.mathematik.tu-darmstadt.de/?id=84&evsid=23&evsver=803>
 - Integrated course: lectures and computer exercises
 - Exercises are treated in groups up to 3 students
 - One (or more) homework assignments to pass this course
 - **MOST IMPORTANT: In order to be able to participate in the exercises, you need account and password in the computer science department!**

Organisation



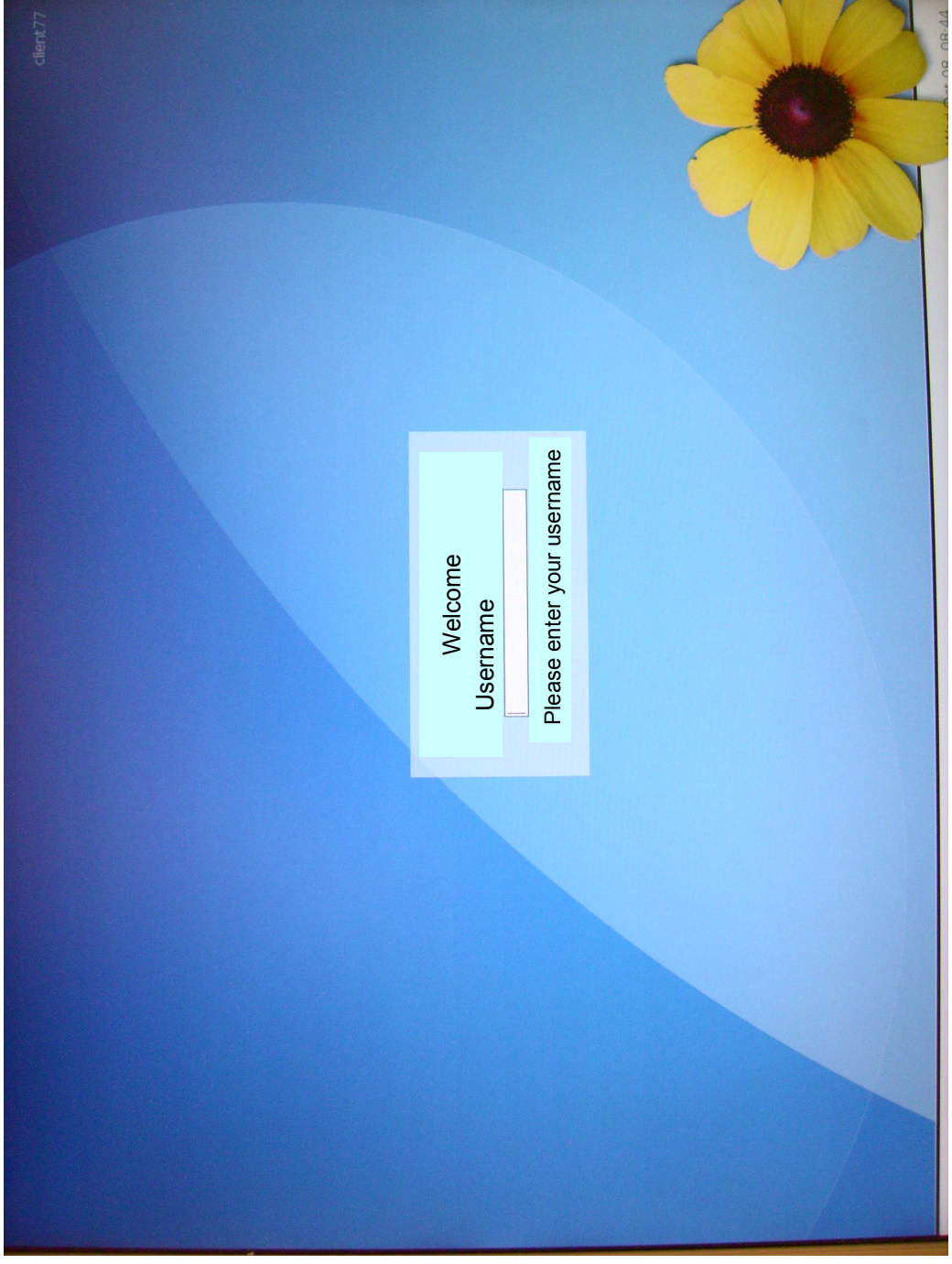
TECHNISCHE
UNIVERSITÄT
DARMSTADT

- How you get an account in the CS department
 - Go to <https://www3.mathematik.tu-darmstadt.de/fb/mathe/lehre-und-studium/elektronisches-veranstaltungssystem.html?evsid=23&evsver=803>
 - Click **Online-Anmeldung**
 - **Deadline is 19th Oct. 2009, 16:00**
You need your HRZ identification (TU-ID) and your Matrikelnummer.
- Your lecturer will give the list of applications to the CS department, and beginning with Oct. 23rd, you can use your account**
 - **Passwords: initial HRZ-account password**

Organisation



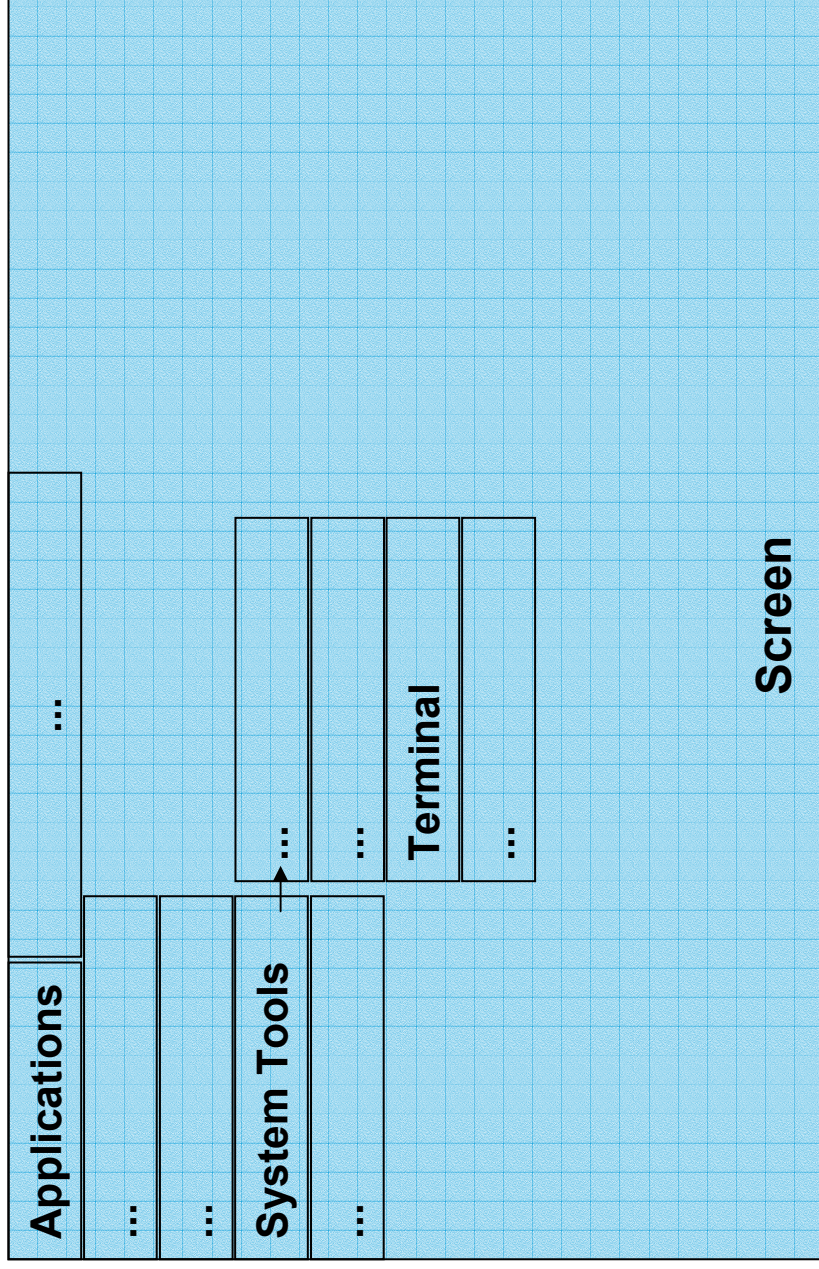
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Organisation



TECHNISCHE
UNIVERSITÄT
DARMSTADT

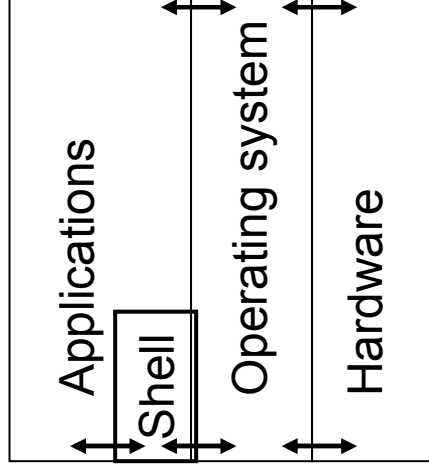


find the programs **terminal** and **kate**

Terminal / Shell



A shell is an interactive program, designed for the man-machine communication. The communication goes via text-commands that are typed in the shell window, and via text output from the machine.



There are commands for:
finding out information about a user, help, finding out the current directory,
listing files and directories, inspecting files, setting your password, modifying
and changing authorizations, copying and moving and removing files, printing ...



Shell, some commands

Some of the most used commands for the shell are the following.

- **pwd**
- **ls**
- **cat <filename>**
- **rm <filename>**
- **man <command>**
- **cd <dirname>**
- **cp <file1> <file2>**
- **mv <file1> <file2>**
- print working directory. Where am I?
- list the files in the current directory
- print a textfile to screen
- remove a file
- manual information o a command
- change the directory to the sub-directory dirname
- copy file1 to file2
- rename the file file1 to the name file2

Why Unix/Linux and not Microsoft-Windows



- Microsoft-Windows is also in use
- Unix is open-source
 - hits the mentality of universities: we do research for the collective good, not for the wealth companies
 - low costs
- Better suited for certain purposes
 - parallel computing
 - multi-user, multi-tasking, remote working more natural

Operating System



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Definition (DIN 44300, translation)

Operating system: Programs of a digital computer system, which build, together with the properties of the system itself, the fundament for the possible operations of the computer system. Especially, these programs operate and monitor the execution of processes.

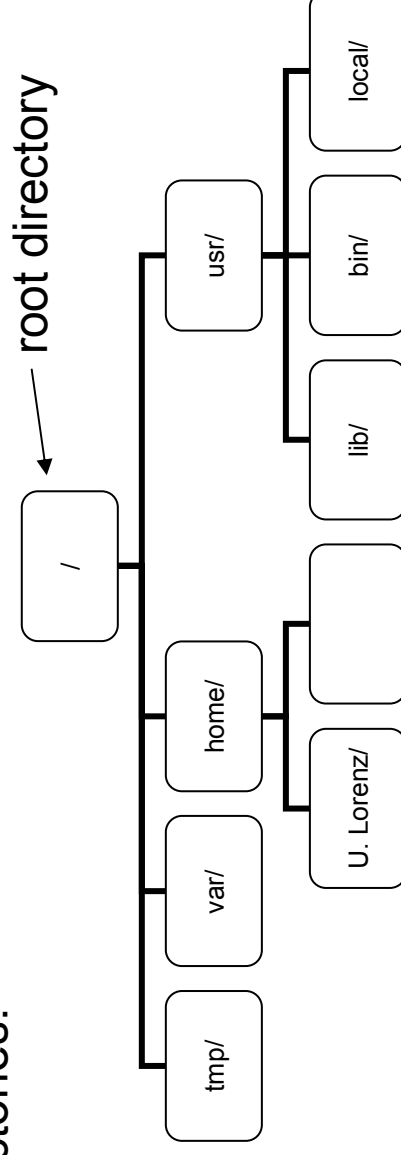
It covers

- File management
- Process management
- Memory management
- Device management

File System



- A file is a sequence of bytes.
- Files are organized in a hierarchy as a tree, with the help of so called directories.



- Structure of other files is unknown to the system. Only defined by user.
- A path of a certain file is the path from the root directory to that file:
e.g. Home directory of U. Lorenz: `/home/U. Lorenz/test.txt`
„test.txt“ therein: `/home/U. Lorenz/test.txt`
- Different files may have the same name in different directories.

C/C++ programming, a short outlook



- Programming language C
 - first examples
 - processing and memory
 - keywords
 - data: variables and their types
 - blocks, statements, and expressions
 - control structures: alternatives, loops, functions, recursion

- Object Orientation with C++
 - classes, objects, dynamic storage allocation
 - encapsulation and inheritance

- Algorithms: e.g. sorting

Literature



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Books:

- *The C++ Programming Language*, by Bjarne Stroustrup
- *Programming in C*, by Kerningham and Ritchie

Links:

- <http://www.cplusplus.com/doc/tutorial>
- <http://www.nongnu.org/c-prog-book/online/index.html>
- <http://www.steveheller.com/cppad/Output/dialogTOC.html>
- <http://www.cppreference.com/wiki/>
- <http://pweb.netcom.com/~tjensen/ptr/pointers.htm>
- <http://www2.its.strath.ac.uk/courses/c/>



Problem, Algorithm and Program

- **Problem**
 - given: the encoding of an input over an alphabet Σ
 - output: the encoding of the answer to a given question
- Example:**
 - given: two natural numbers a and b
 - answer: $\min(a,b)$
- **Algorithm**

is a sequence of pre-defined steps to calculate an answer to a problem
- Example:**
 - „ $z := \min$ imum of a and b “
 - „print z to screen“
- **Program**
 - 1.) have an idea, how the answer of a problem can be generated
 - 2.) express this idea as an algorithm
 - 3.) translate the algorithm into terms such that a stupid machine like a computer is able to execute the algorithm
- Example:**

```
if (a < b) z = a;
else z = b;
printf(„%d\n“, z);
```

C – what’s that? Why C?

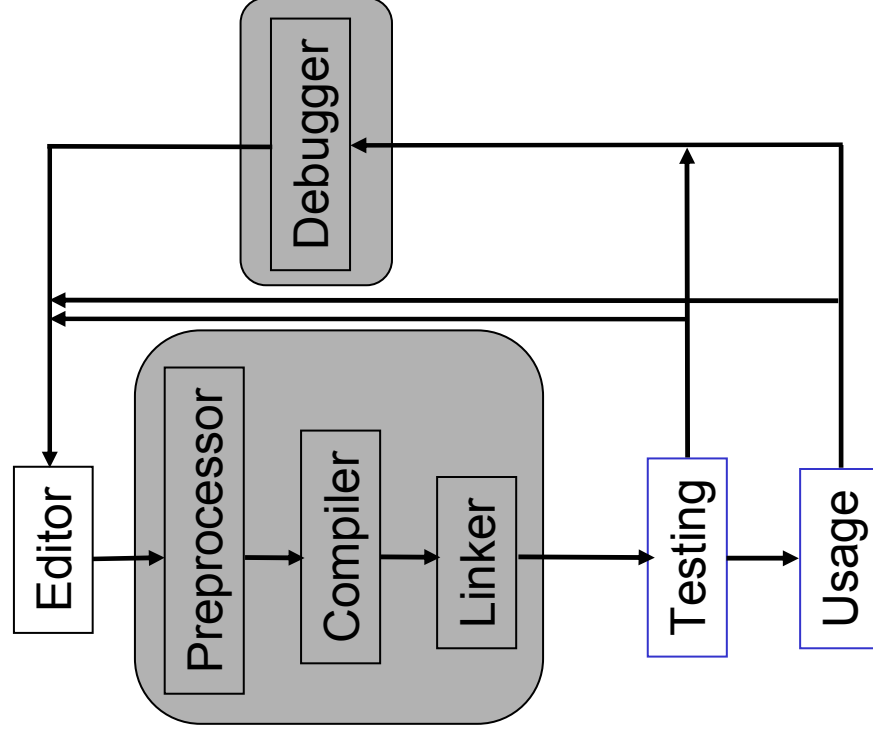


- C is a programming language invented by Brian Kerningham and Dennis Ritchie at Bell Labs in the USA. ~1970
- The language has been standardized in 1989 by ANSI and ISO.
- C is compiled language. This means:
 - instructions are typed into a human readable file with extension .c
 - The C compiler translates these instructions to 0's and 1's that tell a particular CPU chip and Operating System what to do. Creates a file with extension .o
 - These .o files are linked with libraries of other programs that read/write hard-disks, draw characters on the screen etc. Final output is an executable file.
 - this program is run by typing its name or by clicking it
- C-sources are portable to nearly all platforms
- C programs can be very, very fast!
- Operating Systems are written in C (Unix, Windows)
- Applications are written in C (Maple, Word, Cplex, ...) - -
 - Mathematical software often lives from its efficiency.
 - Mathematical software is often dominated by algorithmic aspects.
- For many scientific applications, C has a good ratio of understandability and efficiency

Tools



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Editor

- source files,

Tool: e.g. xemacs: www.xemacs.org
or kate

Preprocessor, Compiler, Linker

- translate the source code of a high level language, given in textfile(s), to machine language, and binds (links) them together

Tool: e.g. gcc, g++

Debugger

- allows to observe the program while it is running.

Tool: e.g. gdb

turnaround cycles in program development

Programming language C



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Example 1:

(minimum program)

```
int main(void)
{ }
```

compilation: gcc small.c -o small

start: ./small

output:

Example 2:

(1) #include <stdio.h>

(2) int main(void) {

(3) printf("Hello World\n");

(4) return 0;

(5) }

Loads an ANSI/ISO standard header file defining the standard C commands to read and write

Declares where the program is to start, that it takes no arguments and that it returns an integer when it stops

Writes *Hello World* to standard output

Tells the program to finish, returning the value 0.

Programming language C



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Example 3:

```
(1) #include <stdio.h>
(2) int main(void) {
(3)     int low=-40, high=140, step=5, f, c;
(4)     c = low;
(5)     while (c<=high) {
(6)         f=32+9*c/5;
(7)         printf("%6d \t %6d \n", c, f);
(8)         c=c+step;
(9)     }
(10)}
```

What does this program do?

Programming language C



Discussion, Example 3:

Line (3) : Variables in C must be **declared** before they can be used.

The keyword **int** defines these variables to be of the type integer (numbers and their types are discussed later). These variables can be **initialized** as in `low=-40`, as soon as they are declared.

The variable declarations are separated by **commas**, and the line is terminated by a **semicolon**. All complete statements in C are terminated by “.”. Characters/letters of variables must be from { 0

1 2 3 4 5 6 7 8 9 _ a b c d e f g h i j k l m n o p q r s t u v w x y z }

Line (4) A C assignment statement. “=” is used for assignment

Line (5) A loop construction

```
while (logical expression) { statements; }
```

Line (6) A C arithmetic statement

Line (7) Print out two numbers of type integer with 6 spaces allocated for each number and a tab separating them.

Line (8) Another C assignment / arithmetic statement.

Programming language C



printf()

- `printf(format string, var1, var2, ..., varN);`

The **format string** is a text template telling C how to print the variables `var1, var2, ..., varN`

Within the format string is:

- plain text that is echoed to the screen
- **Format descriptors** – preceded by a %-sign and followed by a field width number (optional) and a field type letter:
 - **d** – type int
 - **ld** – type long
 - **f** – type float
 - **lf** – type double
 - **Lf** – type long double
 - **c** – type character



Characters

- **character variables** are used for letters
char variables are firstly interpreted as integer numbers between 0 and 127 and then looked up in the so-called ASCII table, in order to interpret that number as a letter.

char(65) is 'A'.

- C recognizes 96 printing ASCII characters (the other characters are called non-printing characters).
One character “\”, is a special character that is used to flag special non-printing characters.

Some of these special characters are

- “\n” – newline
- “\t” – tab
- “\\” – \
- “\” – “
- “\'” – ‘

Programming language C



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Numbers in C

- Integers
 - short value of smallest // largest number
 -2^{15} // $2^{15}-1$
 - int -2^{31} // $2^{31}-1$
 - long -2^{31} // $2^{31}-1$
 - long long -2^{63} // $2^{63}-1$

- Floating Point
 - float absolute value of smallest / largest number
 $\approx 1.401 \times 10^{-45}$ // $\approx 3.403 \times 10^{38}$
 - double $\approx 4.941 \times 10^{-324}$ // $\approx 1.798 \times 10^{308}$
 - long double $\approx 3.645 \times 10^{-4951}$ // $\approx 1.190 \times 10^{4932}$

Arithmetic in C: +, -, *, /

Programming language C



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Example 3:

```
#include <stdio.h>
int main(void) {
    double x=0.7;
    short a=32767, b=-32768, a2, b2;
    int i = 0;
    a2 = a+2;
    b2 = b-2;
    printf("a=%d b=%d a+2=%d b-2=%d\n", a, b, a2, b2);
    while(i < 10) {
        x = 11.0 * x - 7.0;
        printf("%d: %.20lf\n", i, x);
        i=i+1;
    }
}
```