



UNIVERSITÄT ZU LÜBECK  
INSTITUT FÜR MULTIMEDIALE  
UND INTERAKTIVE SYSTEME

# Titel der Arbeit

**Masterarbeit/Bachelorarbeit**

im Rahmen des Studiengangs

**Medieninformatik**

der Universität zu Lübeck

vorgelegt von:

**Vorname Nachname**

ausgegeben und betreut von:

**Titel Name Erstgutachter**

mit Unterstützung von:

**Titel Name Betreuer**

*[Optional:]* Die Arbeit ist im Rahmen einer Tätigkeit bei der Firma XYZ entstanden.

Lübeck, Tag. Monat. Jahr

## Abstract

Der Abstract einer Abschlussarbeit sollte eine kurze Zusammenfassung enthalten, damit der Leser nach einigen Sätzen einen Eindruck davon bekommt, welches Thema bearbeitet wurde. Ein Abstract ist dabei kein “Teaser” sondern eher eine “Executive Summary”.

Dieses Dokument dient als Vorlage und gleichzeitig als kleine Anleitung, um eine Abschlussarbeit mit  $\text{\LaTeX}$  zu erstellen. Um das Template für die eigene Abschlussarbeit zu verwenden, kann einfach der vorhandene Text gelöscht und eigener Text hinzugefügt werden. Das Dokument enthält keine ausführliche Erklärung für das Arbeiten mit  $\text{\LaTeX}$ , da es hierzu eine Vielzahl von Tutorials im Internet gibt. Stattdessen enthält es einige Tipps und Richtlinien. Der Quellcode ist ausführlich dokumentiert, damit es einfach ist das Template für die eigene Arbeit anzupassen.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Related Work . . . . .	1
1.3	Research Questions . . . . .	1
<b>2</b>	<b>Background</b>	<b>2</b>
2.1	Positioning in the distributed Test Support System . . . . .	2
2.2	TCP/IP Stack . . . . .	2
2.3	Linux Network Stack . . . . .	2
2.3.1	Socket Types . . . . .	2
2.3.2	Sending and Receiving a Packet . . . . .	2
2.4	Parameters . . . . .	2
<b>3</b>	<b>Method</b>	<b>3</b>
3.1	Setup . . . . .	4
3.1.1	Hardware Setup . . . . .	4
3.1.2	Software Setup . . . . .	4
3.2	Architecture . . . . .	4
3.2.1	Setup with Ethernet Switch . . . . .	4
3.2.2	Setup with Star Topology . . . . .	4
3.3	TestSuite . . . . .	4
3.3.1	Description of Software Design . . . . .	4
3.3.2	Generation and Measurement of Target Communication . . . . .	4
3.3.3	Recorded and Analyzed Data . . . . .	4
3.4	Generation of additional System Load . . . . .	4
3.4.1	stress-ng . . . . .	4
3.4.2	iPerf . . . . .	4

<b>4</b>	<b>Tutorial zu dieser LaTeX-Vorlage</b>	<b>5</b>
4.1	Verwendung dieser Vorlage . . . . .	5
4.2	Projektstruktur . . . . .	6
4.2.1	Unterkapitel . . . . .	6
4.3	Grafiken . . . . .	6
4.3.1	Vektor- vs Pixelgrafiken . . . . .	6
4.4	Tabellen . . . . .	9
4.5	Quellcode . . . . .	9
4.6	Literatur . . . . .	10
4.7	Abkürzungen . . . . .	12
<b>5</b>	<b>Bibliography</b>	<b>14</b>
<b>A</b>	<b>Appendix</b>	<b>16</b>
<b>B</b>	<b>List of Figures</b>	<b>17</b>
<b>C</b>	<b>List of Tables</b>	<b>18</b>
<b>D</b>	<b>Listings</b>	<b>19</b>

# 1 Introduction

## 1.1 Motivation

## 1.2 Related Work

## 1.3 Research Questions

## 2 Background

### 2.1 Positioning in the distributed Test Support System

### 2.2 TCP/IP Stack

### 2.3 Linux Network Stack

#### 2.3.1 Socket Types

##### 2.3.1.1 UDP Sockets

##### 2.3.1.2 Raw Sockets

##### 2.3.1.3 Packet Sockets

#### 2.3.2 Sending and Receiving a Packet

### 2.4 Parameters



## 3 Method

### 3.1 Setup

#### 3.1.1 Hardware Setup

#### 3.1.2 Software Setup

### 3.2 Architecture

#### 3.2.1 Setup with Ethernet Switch

#### 3.2.2 Setup with Star Topology

### 3.3 TestSuite

#### 3.3.1 Description of Software Design

#### 3.3.2 Generation and Measurement of Target Communication

#### 3.3.3 Recorded and Analyzed Data

### 3.4 Generation of additional System Load

#### 3.4.1 stress-ng

#### 3.4.2 iPerf



## 4 Tutorial zu dieser LaTeX-Vorlage

Dieses Kapitel ist spezifisch für die  $\text{\LaTeX}$ -Vorlage und sollte natürlich in der finalen Abgabe nicht enthalten sein. Dieser Teil der Arbeit kann bei Bedarf durch einen Kommentar einfach ausgeblendet werden (siehe `thesis.tex` Zeile 125).

### 4.1 Verwendung dieser Vorlage

Dieses Template ist für die Verwendung mit `pdflatex` gedacht. Am einfachsten ist es die Vorlage in Overleaf `Overleaf c/o Digital Science, n.d.` zu öffnen. Overleaf ist eine Onlineanwendung zum Arbeiten mit  $\text{\LaTeX}$ , was den Vorteil hat, dass nichts lokal installiert werden muss und mit jedem Betriebssystem gearbeitet werden kann, das über einen Browser verfügt. Außerdem ist es möglich mit mehreren Personen gemeinsam an einem Projekt zu arbeiten. Die Vorlage kann auch mit einer lokalen Installation verwendet werden. Die Website von Overleaf bietet zudem einige gute Tutorials zum Arbeiten mit  $\text{\LaTeX}$ : <https://www.overleaf.com/learn>.

Fehler und Warnungen, die beim Compilieren erzeugt werden, sollten direkt behoben werden, da es später schwierig sein kann den eigentlichen Auslöser einer Fehlermeldung zu finden. Manchmal sieht das Dokument trotz Fehlermeldung oder Warnung korrekt aus, der Fehler macht sich dann aber später bemerkbar. Die Meldungen sind leider oft nicht sehr aussagekräftig, weshalb es am einfachsten ist direkt nach dem Auftreten eines Fehlers den Teil der Arbeit anzuschauen, der als letztes geändert wurde.

## 4.2 Projektstruktur

Der Hauptteil einer Thesis besteht üblicherweise aus mehreren Kapiteln, die verschiedene Aspekte der Arbeit beleuchten. Es ist ratsam für jedes Kapitel eine eigene Tex-Datei anzulegen, damit der Quellcode übersichtlich bleibt. Durch einen numerischen Präfix (z.B.: `20_relatedwork.tex` siehe Abb. 4.1) werden die Quelldateien in der richtigen Reihenfolge in Overleaf angezeigt. Wir verwenden 20 anstatt 2, damit wir nachträglich auch noch 21, 22, etc. einfügen können.

### 4.2.1 Unterkapitel

Unterkapitel sollten ein abgeschlossenes Thema behandeln. Einzelne Unterkapitel in einem Kapitel sind zu vermeiden, also z. B. in Kapitel 2 das Unterkapitel 2.1, aber kein weiteres Unterkapitel. In diesem Fall ist es besser entweder den Inhalt von 2.1 direkt in Kapitel 2 zu schreiben, oder falls 2 und 2.1 thematisch zu weit voneinander entfernt sind, aus Unterkapitel 2.1 ein eigenes Kapitel 3 zu machen. Dieses Unterkapitel ist ein negativ Beispiel dafür.

## 4.3 Grafiken

In der Informatik sind die häufigsten Grafiken entweder Diagramme oder Plots. Beide Arten von Grafiken lassen sich gut als Vektorgrafiken erstellen und einbinden. Der Vorteil von Vektor- gegenüber Pixelgrafiken ist, dass beliebig weit in eine Grafik hereingezoomt werden kann, ohne dass sie unscharf wird. Zudem benötigen Vektorgrafiken meistens weniger Speicherplatz.

### 4.3.1 Vektor- vs Pixelgrafiken

Für die meisten Abbildungen sollte man Vektorgrafiken verwenden, diese können verlustfrei skaliert werden und bieten somit die meisten Freiheiten. Wenn man Grafiken in R erstellt, können die Plots als pdf oder svg-Dateien abgespeichert werden

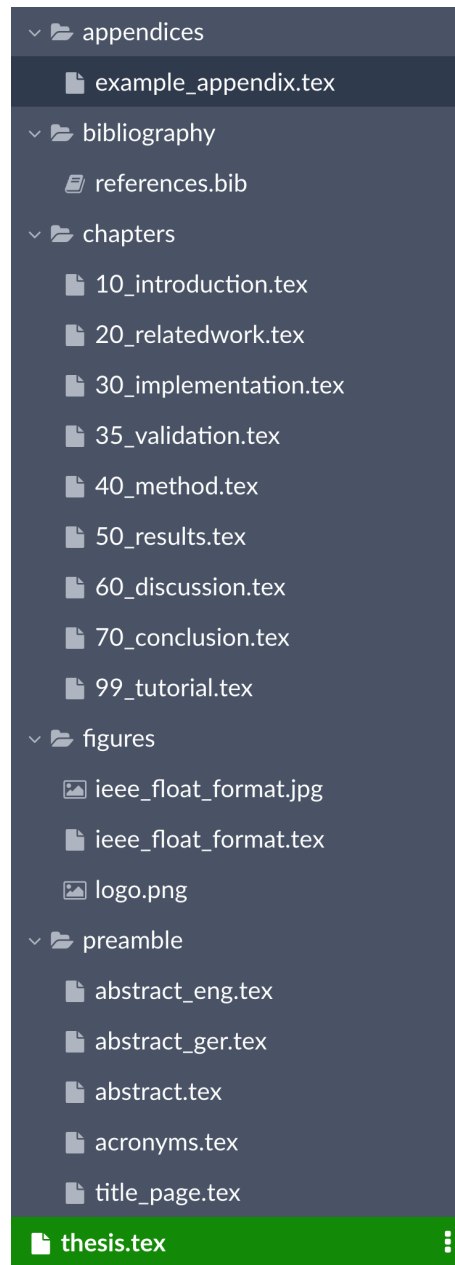


Figure 4.1: Ordnerstruktur eines LaTeX-Projektes

und liegen dann ebenfalls als Vektorgrafik vor. Ein weiteres beliebtes Programm zum Erstellen von Vektorgrafiken ist Inkscape Inkscape Project, 2020. Zudem bieten viele Programme die Möglichkeit eine Grafik z. B. als PDF zu exportieren, was in  $\text{\LaTeX}$  als Vektorgrafik eingebunden werden kann. Adobe Indesign wird auch häufig zum Erstellen von Vektorgrafiken verwendet.

**4.3.1.0.1 Profi-Tipp TikZ.** Grafiken können direkt in  $\text{\LaTeX}$  mit dem TikZ Paket Tantau, 2021 erstellt werden. Die Verwendung ist etwas gewöhnungsbedürftig, da Grafiken mit Code beschrieben werden, bietet aber viele Freiheiten. Außerdem werden die so erstellten Grafiken direkt in  $\text{\LaTeX}$  gerendert und verwenden die selbe Schriftart wie im Text und eine konsistente Schriftgröße im gesamten Dokument.

Pixelgrafiken lassen sich nicht immer vermeiden, z. B. wenn eine Foto in die Arbeit eingebunden werden soll. In diesem Fall sollte darauf geachtet werden, dass die Grafik über eine ausreichende Auflösung verfügt. Eine Auflösung von 300 dpi ist ein guter Richtwert, um beim Drucken ein gutes Ergebnis zu erhalten.

Abbildungen 4.2 und 4.3 zeigen beide den Aufbau des IEEE Floating Point Formats. Abbildung 4.3 ist eine Pixelgrafik, während Abbildung 4.2 mit TikZ erstellt wurde. Der Unterschied wird beim hereinzoomen deutlich.

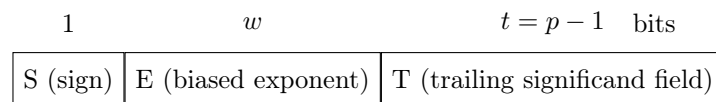


Figure 4.2: Aufbau des IEEE Floating Point Formats als Vektorgrafik mit TikZ erzeugt.

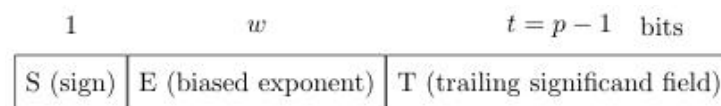


Figure 4.3: Aufbau des IEEE Floating Point Formats als Pixelgrafik.

## 4.4 Tabellen

Tabellen können in L<sup>A</sup>T<sub>E</sub>X direkt erstellt werden. Tabelle 4.1 zeigt ein Beispiel dafür. Einfache Tabellen lassen sich schnell erstellen, bei komplizierteren Tabellen ist es manchmal einfacher zusätzliche Pakete zu verwenden. Mit dem Paket `multirow` können z. B. einfacher Tabellen erstellt werden, bei denen einzelne Zeilen oder Spalten zusammengefasst sind.

Unter <https://www.tablesgenerator.com/> findet man ein hilfreiches online Werkzeug zum erstellen von Tabellen. Wichtig ist es hier den “Default table style” zum “Booktabs table style umzustellen”.

Parameter	binary16	binary32	binary64	binary128
$k$ , storage width in bits	16	32	64	128
$w$ , exponent field width in bits	5	8	11	15
$t$ , significand field width in bits	10	23	52	112
$e_{\max}$ , maximum exponent $e$	15	127	1023	16383
bias, $E - e$	15	127	1023	16383

Table 4.1: IEEE 754-2019 Floating Point Formate als Beispiel für das Einbinden einer Tabelle.

## 4.5 Quellcode

Um Quellcode in die Arbeit einzubinden, können in L<sup>A</sup>T<sub>E</sub>X Listings verwendet werden. Es gibt für populäre Sprachen vorgefertigte Umgebungen, welche die Syntax farblich hervorheben. Quellcode sollte eingebunden werden, wenn eine konkrete Implementierung in einer Sprache erläutert wird. Für die Erklärung eines Algorithmus ist es oft übersichtlicher ein Schaubild oder Pseudocode zu verwenden. Es sollten nur kurze Codeabschnitte eingebunden werden, die für den Leser einfach nachvollziehbar sind und nur den für die Erklärung relevanten Code enthalten. Längere Codeabschnitte können im Anhang stehen. Der komplette Code, der für die Arbeit geschrieben wurde, sollte in einem Repository (Gitlab) abgelegt werden.

Listing 4.1 zeigt ein Beispiel für ein Codelisting in der Programmiersprache C.

Algorithmus 4.1 zeigt einen Routing Algorithmus als Pseudocode. Der Code wurde mit dem Paket algorithm2e Fiorio, 2017 erstellt.

```

1 #include <stdio.h>
2 // comments are highlighted in green
3 void main() {
4     // keywords of the language are highlighted in blue
5     for (int i = 0; i <= 42; ++i) {
6         printf("%d\n", i);
7     }
8     // strings are highlighted in red
9     printf("Hello World!");
10 }

```

Listing 4.1: Beispiel für ein Codelisting in der Sprache C.

```

1 if destination in west direction then
2 |   go West;
3 else if destination in same column then
4 |   if destination in north direction then
5 | |   go North;
6 |   else
7 | |   go South;
8 |   end
9 else if destination in north east direction then
10 |   go North or go East
11 else if destination in south east direction then
12 |   go South or go East
13 else if destination in same row and in east direction then
14 |   go East;
15 else if at destination then
16 |   done;

```

Algorithmus 4.1: West First-Routing Algorithm.

## 4.6 Literatur

Ein Literaturverzeichnis sollte mit dem apa Paket für BibLatex erstellt werden. Dazu wird für jede Quelle ein Eintrag in der Datei `references.bib` angelegt. An der

passenden Stelle im Text können diese Einträge mit dem `\cite{}` Befehl zitiert werden. Für jede Quelle die zitiert wird, legt L<sup>A</sup>T<sub>E</sub>X im Literaturverzeichnis einen Eintrag an.

Beschreibungen der Quellen im Bibtex-Format müssen meistens nicht selbst erstellt werden, sondern können direkt bei vielen Verlagen und Bibliotheken direkt generiert werden. Google bietet mit dem “Scholar-Button” ein Chromium Plugin, mit dem schnell bibtex-Einträge generiert werden können. Bei Google-Scholar generierten Bibtex-Einträgen muss auch eine manuelle Endkontrolle stattfinden, um zu prüfen, ob die Daten in Google korrekt gespeichert waren (z.B. fehlende Autoren, falsche Jahreszahl, etc.).

Es gibt verschiedene Arten wie man mit biblatex zitiert.

`\autocite{valdez2015reducing}` führt zu (Calero Valdez et al., 2015).

`\parencite{valdez2015reducing}` führt zu (Calero Valdez et al., 2015).

`\textcite{valdez2015reducing}` führt zu Calero Valdez et al. (2015).

`\cite{valdez2015reducing}` führt zu Calero Valdez et al., 2015.

`\autocite[siehe auch][S. 13]{valdez2015reducing}` führt zu (siehe auch Calero Valdez et al., 2015, S. 13).

Hier<sup>1</sup> gibt es ein hilfreiches Cheat-Sheet.

Je nach zitierter Dokumentsorte, sieht die Referenz im Literaturverzeichnis anders aus.

- Beispiel für einen Konferenzbeitrag (Nielsen & Molich, 1990)
- Beispiel für einen Journal-Artikel (Hollan et al., 2000)
- Beispiel für ein Buch (Zobel, 2014).
- Beispiel für eine Norm (*Ergonomie der Mensch-System-Interaktion - Teil 220: Prozesse zur Ermöglichung, Durchführung und Bewertung menschenzentrierter Gestaltung für interaktive Systeme in Hersteller- und Betreiberorganisationen*, 2020).

---

<sup>1</sup><https://tug.ctan.org/info/biblatex-cheatsheet/biblatex-cheatsheet.pdf>

- Beispiel für einen Weblink<sup>2</sup>

**Hinweis:** Keine dieser Referenzen müssen Sie in Ihrer Arbeit zitieren!

## 4.7 Abkürzungen

Für jede verwendete Abkürzung kann ein Eintrag in der Datei `acronyms.tex` angelegt werden. Wenn diese Abkürzung im Text zum ersten Mal auftaucht, sollte der Begriff ausgeschreiben werden mit der Abkürzung in Klammern dahinter. Bei weiteren Vorkommen im Text kann dann die eigentliche Abkürzung verwendet werden. In L<sup>A</sup>T<sub>E</sub>X gibt es dafür spezielle Befehle. Beispiel für ausgeschriebene Abkürzung (siehe Quelltext des Dokuments für die entsprechenden Befehle).

Erste Verwendung mit Erläuterung: Network Interface Controller (NIC).

Beispiel für das Verwenden der Abkürzung: NIC.

Die verwendeten Abkürzungen werden automatisch im Abkürzungsverzeichnis aufgelistet.

---

<sup>2</sup><http://imis.uni-luebeck.de> (“Institut für Multimediale und Interaktive Systeme”, 2023)





## 5 Bibliography

- Calero Valdez, A., Brauner, P., Schaar, A. K., Holzinger, A., & Zieffle, M. (2015). Reducing complexity with simplicity-usability methods for industry 4.0. *Proceedings 19th Triennial congress of the IEA*, 9, 14 (cit. on p. 11).
- Hollan, J., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.*, 7(2), 174–196. <https://doi.org/10.1145/353485.353487> (cit. on p. 11).
- Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 249–256. <https://doi.org/10.1145/97243.97281> (cit. on p. 11).
- Zobel, J. (2014). *Writing for computer science*. Springer London. <https://doi.org/10.1007/978-1-4471-6639-9> (cit. on p. 11).

## Online-Quellen

- Fiorio, C. (2017, July 18). *Algorithm2e package* [<https://ctan.space-pro.be/tex-archive/macros/latex/contrib/algorithm2e/doc/algorithm2e.pdf>, accessed on 2022-01-17] (5.3). (Cit. on p. 10).
- Inkscape Project. (2020, April 16). *Inkscape* [<https://inkscape.org>, accessed on 2022-01-13] (0.92.5). (Cit. on p. 8).
- Institut für Multimediale und Interaktive Systeme* [Zugriff am 29.09.2023]. (2023). <https://web.archive.org/web/20230929110531/https://www.imis.uni-luebeck.de/de> (cit. on p. 12).
- Overleaf c/o Digital Science. (n.d.). *Overleaf* [<https://www.overleaf.com/project>, accessed on 2022-01-13]. (Cit. on p. 5).

Tantau, T. (2021, May 15). *The tikz and pgf packages* [<https://github.com/pgf-tikz/pgf>, accessed on 2022-01-13] (3.1.9a). (Cit. on p. 8).

## Normen

*Ergonomie der Mensch-System-Interaktion - Teil 220: Prozesse zur Ermöglichung, Durchführung und Bewertung menschenzentrierter Gestaltung für interaktive Systeme in Hersteller- und Betreiberorganisationen* (Standard). (2020, July). International Organization for Standardization. Geneva, CH. (Cit. on p. 11).

# A Appendix

Der Anhang kann Teile der Arbeit enthalten, die im Hauptteil zu weit führen würden, aber trotzdem für manche Leser interessant sein könnten. Das können z. B. die Ergebnisse weiterer Messungen sein, die im Hauptteil nicht betrachtet werden aber trotzdem durchgeführt wurden. Es ist ebenfalls möglich längere Codeabschnitte anzuhängen. Jedoch sollte der Anhang kein Ersatz für ein Repository sein und nicht einfach den gesamten Code enthalten.

## B List of Figures

4.1	Ordnerstruktur eines LaTeX-Projektes . . . . .	7
4.2	Aufbau des IEEE Floating Point Formats als Vektorgrafik mit TikZ erzeugt. . . . .	8
4.3	Aufbau des IEEE Floating Point Formats als Pixelgrafik. . . . .	8

## C List of Tables

4.1	IEEE 754-2019 Floating Point Formate als Beispiel für das Einbinden einer Tabelle. . . . .	9
-----	--	---

## D Listings

4.1	Beispiel für ein Codelisting in der Sprache C. . . . .	10
-----	--	----