注：此处无需更改。

阅后删除此文本框。



**本科生毕业设计(论文)**

**北京理工大学本科生毕业设计（论文）题目**

**The Subject of Undergraduate Graduation Project (Thesis) of Beijing Institute of Technology**

注：此处是论文中英文题目，中文题目，居中，字体：华文细黑，加黑，字号：二号，行距：多倍行距1.25，间距：段前、段后均为0行，取消网格对齐选项。英文题目，与中文题目对应，居中，字体：Times New Roman，字号：三号，加黑，行距：多倍行距1.25，间距：段前、段后均为0行，取消网格对齐选项。阅后删除此文本框。

|  |  |
| --- | --- |
| 学 院： |  |
| 专 业： |  |
| 学生姓名： |  |
| 学 号： |  |
| 指导教师： |  |

注：此处按照实际情况填写即可。打印（宋体，三号）或手写都可以。阅后删除此文本框。

20XX 年 月 日

**原创性声明**

注：页眉内容无需更改。阅后删除此文本框。

本人郑重声明：所呈交的毕业设计（论文），是本人在指导老师的指导下独立进行研究所取得的成果。除文中已经注明引用的内容外，本文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

特此申明。

本人签名： 日 期： 年 月 日

**关于使用授权的声明**

本人完全了解北京理工大学有关保管、使用毕业设计（论文）的规定，其中包括：①学校有权保管、并向有关部门送交本毕业设计（论文）的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存本毕业设计（论文）；③学校可允许本毕业设计（论文）被查阅或借阅；④学校可以学术交流为目的,复制赠送和交换本毕业设计（论文）；⑤学校可以公布本毕业设计（论文）的全部或部分内容。

本人签名： 日 期： 年 月 日

指导老师签名： 日 期： 年 月 日

**北京理工大学本科生毕业设计（论文）题目**

注：此处是中文题目，居中，字体：黑体，加黑，字号：小二，行距：单倍行距，间距：段前、段后均为1行，取消网格对齐选项。阅后删除此文本框。

摘　要

注：此处无需更改。阅后删除此文本框。

本文……。

摘要正文选用模板中的样式所定义的“正文”，每段落首行缩进2个字符；或者手动设置成每段落首行缩进2个汉字，字体：宋体，字号：小四，行距：固定值22磅，间距：段前、段后均为0行。阅后删除此段。

摘要是一篇具有独立性和完整性的短文，应概括而扼要地反映出本论文的主要内容。包括研究目的、研究方法、研究结果和结论等，特别要突出研究结果和结论。中文摘要力求语言精炼准确，本科生毕业设计（论文）摘要建议300-500字。摘要中不可出现参考文献、图、表、化学结构式、非公知公用的符号和术语。英文摘要与中文摘要的内容应一致。阅后删除此段。

**关键词：北京理工大学；本科生；毕业设计（论文）**

注：此处字体：黑体、小四号、加粗；

一般选3-8个单词或专业术语，且中英文关键词须对应。

阅后删除此文本框。

注：页脚内容为页码，宋体、五号，居中排列。阅后删除此文本框。

**The Subject of Undergraduate Graduation Project (Thesis) of Beijing Institute of Technology**

注：此处是英文题目，居中，字体：Times New Roman，加黑，字号：三号，行距：单倍行距，间距：段前、段后均为1行，取消网格对齐选项。阅后删除此文本框。

Abstract

注：此处无需更改。阅后删除此文本框。

In order to study……

Abstract正文设置成每段落首行缩进2字符，字体：Times New Roman，字号：小四，行距：固定值22磅，间距：段前、段后均为0行。阅后删除此段。

**Key Words: BIT； Undergraduate； Graduation Project (Thesis)**

注：Key Words与摘要正文之间空一行。Key Words与中文“关键词”一致（3-8个）。词间用分号间隔，末尾不加标点；Times New Roman，小四，加粗。

阅后删除此文本框。

目　录

注：在目录页面中点击鼠标右键，选择“更新域”，在弹出窗口中选择“更新整个目录”，确定即可自动生成目录。章、节标题和页码，字体：宋体，字号：小四，不加粗。阅后删除此文本框。

注：此处无需更改。阅后删除此文本框。

[摘　要 I](#_Toc8720747)

[Abstract II](#_Toc8720748)

[第1章 一级题目 1](#_Toc8720749)

[1.1 二级题目 1](#_Toc8720750)

[1.1.1 三级题目 1](#_Toc8720751)

[结　论 3](#_Toc8720752)

[参考文献 4](#_Toc8720753)

[附　录 6](#_Toc8720754)

[致　谢 7](#_Toc8720755)

# 第1章 绪论

1.1 研究背景和意义

在操作系统的发展历程中，任务调度一直是一个重要的课题。在多个任务需要处理时，选择合适的处理顺序、实现高效的任务切换机制，可以提高计算机系统处理任务的性能。

随着计算机技术的发展，任务调度机制的发展呈现出以下几个趋势：一是适应硬件技术的进步，例如从单核任务调度到多核任务调度、从同构多核调度到异构多核调度；二是调度粒度愈发精细，例如调度的对象从进程到线程，之后又出现了比操作系统线程更精简的调度对象，称为协程/轻量级线程/用户级线程（下文中统称协程）。三是调度机制的目标更加专门化，例如应用于嵌入式实时系统的任务调度机制需要尽量缩短任务的反应时间，而应用于高性能服务器的调度机制则要求高吞吐量和低时延。

这其中，以协程为单位的调度往往出现在进程和线程都无法满足性能要求的高并发情景，因此具有比较苛刻的性能要求。同时因为其对任务的划分比较精细，通常具有较高的调度频率。因此，如果能减小每次调度时，任务选择和切换的开销，可以明显提升系统的性能。本课题的目标是实现一种基于硬件的协程调度器，它在保留了中心化调度的优势的前提下，一方面可以取消原有的调度器线程，使更多线程用于处理任务；另一方面可以利用硬件来加速任务的调度，从而提升高频率协程调度场景下的任务处理性能。

中断是许多实时、分时系统提供的机制，其可以使操作系统感知时间的变化并作出反应（时钟中断），可以及时响应外部设备的事件（外部中断），也可以作为多个处理器核心间通信的手段（核间中断）。不过，中断机制与协程调度并不完全适配，其原因是：目前的协程调度机制大多在用户态实现，无法被操作系统感知。而中断的接收和处理都由操作系统完成。在中断处理的过程中，任务以操作系统线程的方式被打断和恢复，切换开销与线程相近，而远高于协程。本课题将协程调度引入操作系统，将其与中断处理更紧密地结合，使得中断处理过程中的上下文切换也可以享受协程机制带来的低开销。

Rust语言是一门适合系统编程的新兴编程语言，其拥有高于C语言的内存安全性、更现代的语法特性和等同C语言的性能。它从语言层面上支持协程，大大方便了协程相关代码的编写。RISC-V指令集则是一套开源、简洁、模块化的指令集。因此，基于Rust语言和RISC-V指令集的操作系统是操作系统开发的新兴方向之一。这一方向上已经出现了一些教学和科研目的的操作系统，例如rCore教学操作系统[1（开题报告）]。因此，本课题以Rust语言和RISC-V指令集作为软硬件平台，可以丰富这一方向的软硬件生态。

1.2 国内外研究现状

本文的研究基于已有的两个研究方向：一是对任务调度机制的研究，二是对中断机制的研究。

1.2.1 任务调度机制

好的任务调度器，需要同时满足两个要求：科学的调度策略和低开销的实现机制[2（开题报告）]。因此，调度策略和实现机制就称为了研究任务调度的两大方向。由于本研究注重实现机制方向，因此也重点介绍实现机制方面的研究概况。

在国际上，调度器的实现机制主要有以下研究和改进：George Prekas等人[6]设计了ZygOS系统，专门优化了调度机制与网络协议栈的实现，以达到更高的网络性能。其将网络协议栈与应用程序的执行流分离，通过队列相互通信。通过该方式，空闲的应用程序可以“窃取”其它CPU核心队列中的数据包并处理，使各个核心间的负载更均衡。Kostis Kaffes等人[2（开题报告）]设计了Shinjuku系统，实现了一个以微秒级调度协程的调度器。因为操作系统提供的机制在微秒时间尺度下开销太大，该调度器实现了自己的抢占机制和上下文切换机制。其使用中心化调度，由一个调度线程和多个工作线程组成。中心化调度的优点是，调度器线程知道整个调度系统的信息，因此可以采用更高效的调度算法；并且，不同核心之间的负载均衡也更简单。不过，设置了一个调度器线程也导致真正进行任务处理的线程比系统使用的线程少了一个。Rishabh Iyer等人[4（开题报告）]在这个思路上改进：一方面，采用协作式调度代替和近似抢占式调度，并使用编译器代码插桩的方式强制应用程序主动让权，在降低上下文切换开销的同时，避免了协作式调度中，占据CPU时间过长的应用程序无法被抢占的问题；另一方面，设置二级队列，在调度器线程的全局队列外设置每个CPU核心的局部队列，降低了工作线程等待调度器线程分配任务的时间。它同时让调度线程在空闲时也处理任务，降低了使用调度线程对效率的影响，然而无法完全消除。

在国内，主要有以下研究和改进：曾素华等人[5（开题报告）]在OSEK操作系统中，证明当低优先级的任务被高优先级的基本任务抢占时，两个任务的执行时序相当于串行执行。因此，他们将这种情况下的抢占转化为函数调用，从而减少上下文切换的开销。钱宏文等人[6（开题报告）]引入FPGA硬件辅助CPU计算，一方面发挥FPGA的可重构优势，将系统的FPGA资源划分为不同的资源块，从而支持多个进程的并行运行；另一方面，改进进程调度机制，将一部分进程分配到FPGA上运行。其具有与软件调度统一的接口，使得用户程序不需修改即可在该系统上运行。该方法创新之处在于使用FPGA作为协处理器，研究和实现了将进程在CPU和协处理器上调度的策略；不足在于将进程传递到FPGA需要进行bit流形式的传输，可能影响效率。尹震宇[7（开题报告）]等人使用硬件实现线程的切换，并为此建立数学模型，设计并实现了一种更适合硬件线程的DR-EDF调度算法。在该方法中，线程的调度和切换过程均可以在硬件中实现，大幅降低了线程切换的开销，因此可以通过更频繁的调度达到更短的响应时间。不过，协程这一更细粒度、更轻量的调度单位比线程更能利用硬件机制提供的低切换开销和高调度频率。

1.2.2 中断机制

如前文所述，时钟中断、外部中断、核间中断这三种中断里，本课题主要关注外部中断。外部中断往往会先由特定的中断芯片接收和汇总，再发往CPU进行处理。已有许多研究通过设计自定义的中断芯片，以改进外部中断的处理过程，从而优化特定指标上的性能。本课题的研究也基于这一方向。

在国际上，主要有以下研究改进：Jerry D. Erwin等人的研究[1]使中断芯片可以动态改变各个中断源的优先级以适应外部环境的变化。其将到来的中断放入优先级队列，并只在队列存在优先级比CPU当前处理的优先级更高的中断时，将对应的中断通知CPU。不过，该设计以一组优先级决定中断接受和加入队列的顺序，以另一组优先级决定中断通知CPU的顺序，分裂的优先级设置可能导致高优先级的中断因为入队优先级低而无法及时处理。Jupyung Lee等人[3（开题报告）]改进了中断处理机制：一方面将中断处理函数的执行移到目标进程中，减少了上下文切换的次数；另一方面，建立中断与进程间的关联，从而将最短的中断处理路径留给优先级最高的中断-进程。通过这些机制，降低了高优先级任务的中断开销。Fabian Scheler等人[2]则为中断芯片内部、中断芯片和CPU之间设置了统一的优先级，解决了高优先级的任务会被低优先级的中断处理程序抢占（优先级倒置）的问题。它在中断到来时，直接将对应的处理程序放入内存中的调度器数据结构，而只在中断优先级高于CPU运行的任务优先级时，才会打断CPU当前的任务。这一思路将中断处理的上下文切换转化为普通任务的上下文切换，对中断与协程调度的融合具有指导意义：如果将中断处理程序与一般任务都实现为协程，就能以协程的切换开销实现中断处理的上下文切换。

在国内，主要有以下研究改进：杨媛媛等人[3]在软件层面设置处理队列并按先入先出顺序处理其中的任务，从而精简中断处理程序，使其不需执行真正的中断处理逻辑，而只需将处理逻辑所需参数加入处理队列即可退出。中断处理程序的精简可以减少中断嵌套的发生，缓解其带来的处理超时、资源抢占等问题。然而，如果先到来的低优先级中断已经入队，后到来的高优先级中断就无法抢占，只能按照队列的先入先出顺序，等待低优先级中断对应的任务执行完成后再执行，从而延长了高优先级任务的响应时间。舒生亮等人[4]为Matrix DSP平台设计和实现了专用的中断处理器，其可以接收各种来源的中断并为它们分配优先级，使CPU响应优先级最高的、使能的中断。其设计专门适配了平台CPU的流水线机制，使得进入中断处理例程延时开销降低到仅4个CPU周期。张旭等人[6]在Linux操作系统上实现了一个用户态任务调度框架，其在用户态实现了任务调度、高精度时钟和中断处理的功能，并且性能均优于Linux内核的实现。任务调度方面，该系统申请的处理器线程中，一个线程用于运行定时器，其余线程均用于运行任务，且各个线程具有独立的任务调度数据结构。系统使用位图算法进行高性能的任务调度。中断处理方面，该系统在内核的中断处理程序中构建快速跳转栈，实现中断在用户态的调度框架内的处理。进入用户态后，再使用软中断机制，将较长的处理逻辑作为任务创建，保证中断处理函数的快速退出。该研究实现了一个功能较为完整的调度系统，对本文的研究有重要的参考价值。与其基于Linux系统相比，本文的研究基于结构更加简单的软硬件系统，因此可以进一步简化部分机制的实现。

1.3 研究内容和关键问题

1.3.1 研究主要内容

本研究分为原型设计和对比测试两部分。首先在QEMU模拟器中设计并实现基于硬件的共享调度器，并验证其正确性；之后在ArceOS，一个基于Rust语言、支持RISC-V指令集的操作系统中，添加基于该调度器的协程支持和外部中断支持，实现异步任务调度和网络性能优化；最后，将其和现有的任务调度机制进行性能测试，对比分析测试结果。

本研究使用QEMU模拟器实现一个模拟硬件设备，其具有中断处理和任务调度两个功能。它可以接收外部中断，并将中断向量表中对应的中断处理协程作为一个高优先级任务加入调度器中进行调度，以实现异步处理中断的效果，减少因中断处理而产生的切换开销。它维护一组不同优先级的先入先出队列，以实现协程调度的功能。软件向其中放入就绪的协程，并从中取出下一个运行的协程，其为最高优先级的非空队列中的最早加入的协程。

本研究修改了ArceOS的任务调度模块，使其能够利用我们实现的硬件设备进行任务调度和中断处理。修改后的任务调度模块可以统一地调度线程与协程，从而既能利用协程的低切换开销，又能兼容ArceOS原有的设计，使得适用于该系统的应用程序不加修改即可运行。其能以线程或协程的形式注册外部中断处理例程，从而在外部中断到来时，加入调度器中运行。

本研究在此基础上修改ArceOS的网络模块，利用新增的协程调度功能和外部中断机制，优化数据包发送和接收的处理流程。其采用两种优化方式，并比较优化的效果：一是创建专门的轮询任务，不断轮询网卡和网络协议栈；二是在接收到中断时，才调用一次相应的轮询函数。

本研究之后对修改后的软硬件系统进行性能测试，同时与未修改的系统进行比较。测试结果表明，在低负载情况下，修改后的系统与修改前性能相差不大；在高负载情况下，修改后的系统优于修改前。

1.3.2关键问题

本项研究的协程调度器具有中断处理和协程调度两个功能，其分别解决两方面的关键问题：

在中断处理方面，传统的中断处理机制需要抢占处理器现有的执行流，这会带来上下文保存和恢复的较大时间开销。本项研究通过将外部中断的处理转化为协程调度，可以利用协程机制降低上下文切换的开销，提高任务处理效率。该方法可以运用在一些对实时性要求不高的外部中断上。

在协程调度方面，协程调度频率较高，软件实现的调度器会产生较大的开销。同时，软件实现的调度机制需要一个线程来运行调度器本身，这减少了可以处理工作负载的线程数。本研究拟采用硬件实现调度器的方案来解决这一关键问题，尽量降低任务切换和中断处理的CPU资源开销和时间开销，同时使更大比例的CPU资源用于处理工作负载。

1.4 全文组织结构

本文的第1章介绍简要介绍了研究背景和国内外研究现状，之后引出本文研究的内容，以及解决的关键问题。第2章对本项研究涉及的技术做简单介绍，包括任务调度的机制和分类，以及Rust语言中的协程。

本文的第3章描述了本研究实现的系统的设计，包括总体架构、QEMU模拟硬件的设计、操作系统层次的设计。第4章介绍QEMU模拟器与本项研究相关的内容，并详细说明了本项研究的模拟硬件部分在QEMU模拟器中的实现。第5章介绍ArceOS操作系统与本项研究相关的内容，并详细说明了本项研究的软件部分在该操作系统中的实现。

本文的第6章描述本项研究中开展的实验评估。其介绍了实验环境和各个实验的设计。之后，给出了各个实验的结果，分析导致该结果的原因。

本文的第7章总结了全文的内容，分析优点与不足，展望了本研究在未来可能的改进方向。

# 第1章 一级题目

1.1 二级题目

1.1.1 三级题目

正文……

正文部分：宋体、小四；正文行距：22磅；间距段前段后均为0行。阅后删除此段。

图、表居中，图注标在图下方，表头标在表上方，宋体、五号、居中，1.25倍行距，间距段前段后均为0行，图表与上下文之间各空一行。阅后删除此段。

图-示例：（阅后删除此段）



图1-1 标题序号

表-示例：（阅后删除此段）

表1-1 统计表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 产量 | 销量 | 产值 | 比重 |
| 手机 | 1000 | 10000 | 500 | 50% |
| 计算机 | 5500 | 5000 | 220 | 22% |
| 笔记本电脑 | 1100 | 1000 | 280 | 28% |
| 合计 | 17600 | 16000 | 1000 | 100% |

注：正文页脚内容为页码，宋体、五号，居中排列。阅后删除此文本框。

公式标注应于该公式所在行的最右侧。对于较长的公式只可在符号处（+、-、\*、/、≤≥等）转行。在文中引用公式时，在标号前加“式”，如式（1-2）。阅后删除此段。

公式-示例：（阅后删除此段）

 (1-1)

注：此处无需更改。阅后删除此文本框。

结　论

本文结论……。

结论作为毕业设计（论文）正文的最后部分单独排写，但不加章号。结论是对整个论文主要结果的总结。在结论中应明确指出本研究的创新点，对其应用前景和社会、经济价值等加以预测和评价，并指出今后进一步在本研究方向进行研究工作的展望与设想。结论部分的撰写应简明扼要，突出创新性。阅后删除此段。

结论正文样式与文章正文相同：宋体、小四；行距：22磅；间距段前段后均为0行。阅后删除此段。

参考文献

[1] Jerry D. Erwin and E. Douglas Jensen. 1970. Interrupt processing with queued content-addressable memories. In Proceedings of the November 17-19, 1970, fall joint computer conference (AFIPS '70 (Fall)). Association for Computing Machinery, New York, NY, USA, 621–627. <https://doi.org/10.1145/1478462.1478553>

[2] Fabian Scheler, Wanja Hofer, Benjamin Oechslein, Rudi Pfister, Wolfgang Schröder-Preikschat, and Daniel Lohmann. 2009. Parallel, hardware-supported interrupt handling in an event-triggered real-time operating system. In Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems (CASES '09). Association for Computing Machinery, New York, NY, USA, 167–174. <https://doi.org/10.1145/1629395.1629419>

[3] 杨媛媛,王晓华,李敏,等. 基于Xilinx FPGA的中断处理[J]. 电脑知识与技术,2021,17(5):244-245.

[4] 舒生亮,孙永节,万江华. Matrix DSP中断处理系统的设计与实现[J]. 计算机工程与科学,2012,34(1):64-68. DOI:10.3969/j.issn.1007-130X.2012.01.011.

[5] 张旭,顾乃杰,苏俊杰.一种Linux用户态实时多任务调度框架[J].中国科学技术大学学报,2017,47(08):635-643.

[6] George Prekas, Marios Kogias, and Edouard Bugnion. 2017. ZygOS: Achieving Low Tail Latency for Microsecond-scale Networked Tasks. In Proceedings of the 26th Symposium on Operating Systems Principles (SOSP '17). Association for Computing Machinery, New York, NY, USA, 325–341. https://doi.org/10.1145/3132747.3132780

注：此部分蓝色字体为注释，阅后可删除；黑色字体为具体示例。

阅后删除此文本框。

注：此处无需更改。阅后删除此文本框。

参考文献

参考文献书写规范

参考国家标准《信息与文献参考文献著录规则》【GB/T 7714—2015】，参考文献书写规范如下：

**1. 文献类型和标识代码**

普通图书：M 会议录：C 汇编：G 报纸：N

期刊：J 学位论文：D 报告：R 标准：S

专利：P 数据库：DB 计算机程序：CP 电子公告：EB

档案：A 舆图：CM 数据集：DS 其他：Z

**2. 不同类别文献书写规范要求**

**期刊**

[序号]主要责任者. 文献题名[J]. 刊名, 出版年份, 卷号(期号): 起止页码.

[1] 余雄庆. 飞机总体多学科设计优化的现状与发展方向[J]. 南京航空航天大学学报, 2008,40(4): 417-426.

[2] Hajela P., Bloebaumj C. L., Sobieszczanski-Sobieski J. Application of Global Sensitivity Equations in Multidisciplinary Aircraft Synthesis[J]. Journal of Aircraft, 1990, 27(12):1002-110.

**普通图书**

[序号]主要责任者. 文献题名[M]. 出版地: 出版者, 出版年. 起止页码.

[3] 李成智,李小宁,田大山. 飞行之梦—航空航天发展史概论[M]. 北京: 北京航空航天大学, 2004.

[4] Raymer D. P. Aircraft Design: A Conceptual Approach[M]. Reston, Virginia: American Institute of Aeronautics and Astronautics, Inc., 2006.

**会议论文集**

[序号]析出责任者. 析出题名[A]. 见(英文用In): 主编. 论文集名[C]. (供选择项: 会议名, 会址, 开会年)出版地: 出版者, 出版年. 起止页码.

[5] 孙品一. 高校学报编辑工作现代化特征[A]. 见: 张为民编. 中国高等学校自然科学学报研究会. 科技编辑学论文集(2)[C]. 北京: 北京师范大学出版社, 1998. 10-22.

**专著中析出的文献**

[序号]析出责任者. 析出题名[A]. 见(英文用In): 专著责任者. 书名[M]. 出版地: 出版者, 出版年.起止页码.

[6] 罗云. 安全科学理论体系的发展及趋势探讨[A]. 见: 白春华,何学秋,吴宗之. 21世纪安全科学与技术的发展趋势[M]. 北京: 科学出版社, 2000. 1-5.

**学位论文**

[序号]主要责任者. 文献题名[D]. 保存地: 保存单位, 年份.

[7] 张和生. 嵌入式单片机系统设计[D]. 北京: 北京理工大学, 1998.

[8] Sobieski I. P. Multidisciplinary Design Using Collaborative Optimization[D]. United States -- California: Stanford University, 1998.

**报告**

[序号]主要责任者. 文献题名[R]. 报告地: 报告会主办单位, 年份.

[9] 冯西桥. 核反应堆压力容器的LBB分析[R]. 北京: 清华大学核能技术设计研究院, 1997.

[10] Sobieszczanski-Sobieski J. Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems[R], NASA CP-3031, 1989.

**专利文献**

[序号]专利所有者. 专利题名[P]. 专利国别: 专利号, 发布日期.

[11] 姜锡洲. 一种温热外敷药制备方案[P]. 中国专利: 881056078, 1983-08-12.

**国际、国家标准**

[序号]标准代号. 标准名称[S]. 出版地: 出版者, 出版年.

[12] GB/T 16159—1996. 汉语拼音正词法基本规则[S]. 北京: 中国标准出版社, 1996.

**报纸文章**

[序号]主要责任者. 文献题名[N]. 报纸名, 出版年, 月(日): 版次.

[13] 谢希德. 创造学习的思路[N]. 人民日报, 1998, 12(25): 10.

**电子文献**

[序号]主要责任者. 电子文献题名[文献类型/载体类型]. 电子文献的出版或可获得地址(电子文献地址用文字表述), 发表或更新日期/引用日期(任选).

[14]姚伯元. 毕业设计(论文)规范化管理与培养学生综合素质[EB/OL]. 中国高等教育网教学研究, 2005-2-2.

关于参考文献的未尽事项可参考国家标准《信息与文献参考文献著录规则》（GB/T 7714—2015）

注：此处无需更改。阅后删除此文本框。

附　录

附录相关内容…

附录是毕业设计（论文）主体的补充项目，为了体现整篇文章的完整性，写入正文又可能有损于论文的条理性、逻辑性和精炼性，这些材料可以写入附录段，但对于每一篇文章并不是必须的。附录依次用大写正体英文字母A、B、C……编序号，如附录A、附录B。阅后删除此段。

附录正文样式与文章正文相同：宋体、小四；行距：22磅；间距段前段后均为0行。阅后删除此段。

注：附录和致谢页脚内容为页码，宋体、五号，居中排列。阅后删除此文本框。

注：此处无需更改。阅后删除此文本框。

致　谢

值此论文完成之际，首先向我的导师……

致谢正文样式与文章正文相同：宋体、小四；行距：22磅；间距段前段后均为0行。阅后删除此段。