|  |  |
| --- | --- |
|  |  |
|  |  |

Diagram

Description automatically generated

程序调试实践

Program Debug Practice

王浩

（软件工程师）

公司：安波福电子（苏州）有限公司

部门：AS&UX CAM&TEL TEAM

邮箱：luis.wang@aptiv.com

导师：

日期：2022/10/30

**安波福电子**

**论文原创性声明**

本人郑重声明：所提交的论文是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含其他个人或集体已经发表或撰写过的研究成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。

论文作者签名： 年 月 日

程序调试实践

中文摘要

随着计算机行业的不断发展和信息技术的继续迭代，越来越多的行业在计算机和互联网的帮助下来提升自身的效率。这对于软件工程来说意味着会有更复杂更多样的需求，而且这些需求也会使得代码变得复杂，程序更加庞大。而对于程序员们来说，这些变化意味着他们需要有更优秀的编程能力和代码调试能力。

本文主要介绍当程序出了不符合预期的表现时，也就是当程序出了BUG时，程序员应当如何去找到出现这个问题的原因。当然为了解释清楚BUG的来龙去脉，本文来将会介绍与之有关的计算机的必要知识，这些知识主要集中在操作系统和程序计算机中的如何编译运行的。

由于目前开发的环境大多集中于UNIX体系的操作系统中，所以本文着重介绍的还是在Linux的操作系统下程序调试相关内容，其中最主要涉及的就是Linux下关于调试方法的研究，比如GDB等调试工具的使用和在调试过程的中应用等。

**关键词：软件调试、Linux、GDB、BUG、DEBUG**

**Program Debug Practice**

**Abstract**

With the development of computer industry and information technology grows, more and more industries are making themselves more efficient with the help of internet and computer. it means that there are more complex and diversified requirement for software engineering, which will make code become sophisticated and make program become huge. Meanwhile, these changes will require programmer with perfect coding skills and excellent code debug ability to cover these solutions.

This paper mainly introduces how to find out the cause of problem for programmer when the program appears non-expected performance, in other words, the program appears with a bug. Of course, in order to better explain the context of the bug, this paper also will introduce some related necessary knowledge about computer and program, and this knowledge mainly center on operation system and how does the program compile and run in the computer.

So far, because of the development environment mainly fasten-on class of the UNIX operation system, so this article focuses on the introduction of the relevant content of program debug technique under Linux operation system, in which the most of important of all is about research on debugging methods in Linux, such as the use of debugging tools and application during debugging process.

**Keywords:** **software、Linux、GDB、BUG、DEBUG**

Table of Contents

[第一章 绪 论 1](#_Toc93417709)

[1.1 程序调试研究的目的 1](#_Toc93417710)

[1.2 程序调试研究的意义 1](#_Toc93417711)

[第二章 程序调试的介绍 1](#_Toc93417712)

[2.1 程序调试的背景 1](#_Toc93417713)

[2.1.1 什么是程序调试 1](#_Toc93417714)

[2.1.2 什么是BUG 1](#_Toc93417715)

[2.1.3 程序调试和程序测试的区别 1](#_Toc93417716)

[2.2 程序调试的目的 1](#_Toc93417717)

[2.2.1 为什么要进行程序调试 1](#_Toc93417718)

[2.2.2 什么时候进行程序调试 1](#_Toc93417719)

[2.2.3 程序调试的步骤 1](#_Toc93417720)

[第三章 调试前的知识准备 1](#_Toc93417721)

[3.1 操作系统相关知识 1](#_Toc93417722)

[3.1.1 字节序 1](#_Toc93417723)

[3.1.2 寄存器 1](#_Toc93417724)

[3.1.3 地址 1](#_Toc93417725)

[3.1.4 数据类型 1](#_Toc93417726)

[3.1.5 堆栈 1](#_Toc93417727)

[3.2 程序运行相关知识 2](#_Toc93417728)

[3.2.1 程序在计算机中的存储结构 2](#_Toc93417729)

[3.2.2 程序的编译链接过程 2](#_Toc93417730)

[3.2.3 程序的运行过程 2](#_Toc93417731)

[3.3 汇编语言相关知识 2](#_Toc93417732)

[3.3.1 汇编语言的常用语法 2](#_Toc93417733)

[3.3.2 汇编语言的使用分析 2](#_Toc93417734)

[第四章 调试的知识准备 2](#_Toc93417735)

[4.1 调试平台差异介绍 2](#_Toc93417736)

[4.2 调试工具介绍 2](#_Toc93417737)

[4.3 GDB基础命令 2](#_Toc93417738)

[4.4 GDB调试步骤 2](#_Toc93417739)

[第五章 简单的程序调试实践 3](#_Toc93417740)

[5.1 简单的main程序调试实践 3](#_Toc93417741)

[5.2 带有函数的程序调试实践 3](#_Toc93417742)

[5.3 加载静态库的程序调试实践 3](#_Toc93417743)

[5.4 加载动态库的程序调试实践 3](#_Toc93417744)

[第六章 多进程程序调试实践 3](#_Toc93417745)

[6.1 多进程程序编译运行 3](#_Toc93417746)

[6.2 多进程程序调试实践 3](#_Toc93417747)

[第七章 多线程程序调试实践 3](#_Toc93417748)

[7.1 多线程程序调试实践 3](#_Toc93417749)

[7.2 多线程程序调试实践 3](#_Toc93417750)

[第八章 内核有关的调试实践 3](#_Toc93417751)

[8.1 与程序崩溃有关的调试实践 3](#_Toc93417752)

[8.2 与程序停止响应的调试实践 3](#_Toc93417753)

[8.3 与系统运行缓慢的调试实践 3](#_Toc93417754)

[8.4 与CPU负载过高的调试实践 3](#_Toc93417755)

[第九章 总结 3](#_Toc93417756)

[参考文献 4](#_Toc93417757)

[致谢 4](#_Toc93417758)

# 绪 论

## 程序调试研究的目的

这篇论文主要是进行程序调试的研究，通过对程序调试的原因，目的，方法的探索和介绍，达到当程序出现不符合预期的现象时，程序员能快速正确的定位到原因所在，并且能解决问题。

随着现在互联网和各行各业的深度结合，数字化的趋势越来越明显，程序员需要的编码能力和精准的debug能力也需要水涨船高。为了更清楚的弄清楚程序出现问题的根本原因和机制，所以就有这篇论文研究的出现。

## 程序调试研究的意义

通过对程序调试的研究，可以使得程序员更好的熟悉调试的方法步骤，更好的理解bug的出现原因和debug的方法，培养程序员在面对出现BUG时解决和分析能力。最终可以写出更规范，更安全的代码。

# 程序调试的介绍

## 2.1 程序调试的背景

每一个程序都是为了解决一个问题。所以在一个完整的，规范的项目实现过程中，程序的最初来源是需求定义，然后是根据需求，进行程序的总体设计，然后再是根据总体设计制定详细设计，再经过程序员的编码，系统测试步骤，程序测试调试等步骤。当然这些步骤的能实现的前提是项目立项，需求确定。

而本文主要研究的是程序在编码之后以及经过测试工程师测试之后出现过问题后的调试过程。

程序调试的目的就是为了解决程序中不符合预期现象。

### 2.1.1 什么是程序调试

### 2.1.2 什么是BUG

### 2.1.3 程序调试和程序测试的区别

## 2.2 程序调试的目的

### 2.2.1 为什么要进行程序调试

### 2.2.2 什么时候进行程序调试

### 2.2.3 程序调试的步骤

# 调试前的知识准备

## 3.1 操作系统相关知识

### 3.1.1 字节序

### 3.1.2 寄存器

### 3.1.3 地址

### 3.1.4 数据类型

### 3.1.5 堆栈

## 3.2 程序运行相关知识

### 3.2.1 程序在计算机中的存储结构

### 3.2.2 程序的编译链接过程

### 3.2.3 程序的运行过程

## 3.3 汇编语言相关知识

### 3.3.1 汇编语言的常用语法

### 3.3.2 汇编语言的使用分析

# 调试的知识准备

### 4.1 操作系统介绍

### 4.2 调试工具介绍

### 4.3 GDB基础命令

### 4.4 GDB调试步骤

# 简单的程序调试实践

### 5.1 简单的main程序调试实践

### 5.2 带有函数的程序调试实践

### 5.3 加载静态库的程序调试实践

### 5.4 加载动态库的程序调试实践

### 5.5 总结

# 多进程程序调试实践

### 6.1 多进程程序编译运行

### 6.2 多进程程序调试实践

# 多线程程序调试实践

### 7.1 多线程程序编译运行

### 7.2 多线程程序调试实践

# 内核有关的调试实践

### 8.1 与程序崩溃有关的调试实践

### 8.2 与程序停止响应的调试实践

### 8.3 与系统运行缓慢的调试实践

### 8.4 与CPU负载过高的调试实践

# 总结

# 参考文献

# 致谢