

**计算机系统**

**大作业**

题 目 程序人生-Hello’s P2P

专 业 计算机类

学　　 号 1190200523

班　　 级 1903002

学 生 石翔宇

指 导 教 师 郑贵滨

**计算机科学与技术学院**

**2021年6月**

**摘 要**

摘要是论文内容的高度概括，应具有独立性和自含性，即不阅读论文的全文，就能获得必要的信息。摘要应包括本论文的目的、主要内容、方法、成果及其理论与实际意义。摘要中不宜使用公式、结构式、图表和非公知公用的符号与术语，不标注引用文献编号，同时避免将摘要写成目录式的内容介绍。

**关键词：**关键词1；关键词2；……；

**（摘要0分，缺失-1分，根据内容精彩称都酌情加分0-1分）**

**目 录**

[第1章 概述 - 4 -](#_Toc532238396)

[1.1 Hello简介 - 4 -](#_Toc532238397)

[1.2 环境与工具 - 4 -](#_Toc532238398)

[1.3 中间结果 - 4 -](#_Toc532238399)

[1.4 本章小结 - 4 -](#_Toc532238400)

[第2章 预处理 - 5 -](#_Toc532238401)

[2.1 预处理的概念与作用 - 5 -](#_Toc532238402)

[2.2在Ubuntu下预处理的命令 - 5 -](#_Toc532238403)

[2.3 Hello的预处理结果解析 - 5 -](#_Toc532238404)

[2.4 本章小结 - 5 -](#_Toc532238405)

[第3章 编译 - 6 -](#_Toc532238406)

[3.1 编译的概念与作用 - 6 -](#_Toc532238407)

[3.2 在Ubuntu下编译的命令 - 6 -](#_Toc532238408)

[3.3 Hello的编译结果解析 - 6 -](#_Toc532238409)

[3.4 本章小结 - 6 -](#_Toc532238410)

[第4章 汇编 - 7 -](#_Toc532238411)

[4.1 汇编的概念与作用 - 7 -](#_Toc532238412)

[4.2 在Ubuntu下汇编的命令 - 7 -](#_Toc532238413)

[4.3 可重定位目标elf格式 - 7 -](#_Toc532238414)

[4.4 Hello.o的结果解析 - 7 -](#_Toc532238415)

[4.5 本章小结 - 7 -](#_Toc532238416)

[第5章 链接 - 8 -](#_Toc532238417)

[5.1 链接的概念与作用 - 8 -](#_Toc532238418)

[5.2 在Ubuntu下链接的命令 - 8 -](#_Toc532238419)

[5.3 可执行目标文件hello的格式 - 8 -](#_Toc532238420)

[5.4 hello的虚拟地址空间 - 8 -](#_Toc532238421)

[5.5 链接的重定位过程分析 - 8 -](#_Toc532238422)

[5.6 hello的执行流程 - 8 -](#_Toc532238423)

[5.7 Hello的动态链接分析 - 8 -](#_Toc532238424)

[5.8 本章小结 - 9 -](#_Toc532238425)

[第6章 hello进程管理 - 10 -](#_Toc532238426)

[6.1 进程的概念与作用 - 10 -](#_Toc532238427)

[6.2 简述壳Shell-bash的作用与处理流程 - 10 -](#_Toc532238428)

[6.3 Hello的fork进程创建过程 - 10 -](#_Toc532238429)

[6.4 Hello的execve过程 - 10 -](#_Toc532238430)

[6.5 Hello的进程执行 - 10 -](#_Toc532238431)

[6.6 hello的异常与信号处理 - 10 -](#_Toc532238432)

[6.7本章小结 - 10 -](#_Toc532238433)

[第7章 hello的存储管理 - 11 -](#_Toc532238434)

[7.1 hello的存储器地址空间 - 11 -](#_Toc532238435)

[7.2 Intel逻辑地址到线性地址的变换-段式管理 - 11 -](#_Toc532238436)

[7.3 Hello的线性地址到物理地址的变换-页式管理 - 11 -](#_Toc532238437)

[7.4 TLB与四级页表支持下的VA到PA的变换 - 11 -](#_Toc532238438)

[7.5 三级Cache支持下的物理内存访问 - 11 -](#_Toc532238439)

[7.6 hello进程fork时的内存映射 - 11 -](#_Toc532238440)

[7.7 hello进程execve时的内存映射 - 11 -](#_Toc532238441)

[7.8 缺页故障与缺页中断处理 - 11 -](#_Toc532238442)

[7.9动态存储分配管理 - 11 -](#_Toc532238443)

[7.10本章小结 - 12 -](#_Toc532238444)

[第8章 hello的IO管理 - 13 -](#_Toc532238445)

[8.1 Linux的IO设备管理方法 - 13 -](#_Toc532238446)

[8.2 简述Unix IO接口及其函数 - 13 -](#_Toc532238447)

[8.3 printf的实现分析 - 13 -](#_Toc532238448)

[8.4 getchar的实现分析 - 13 -](#_Toc532238449)

[8.5本章小结 - 13 -](#_Toc532238450)

[结论 - 14 -](#_Toc532238451)

[附件 - 15 -](#_Toc532238452)

[参考文献 - 16 -](#_Toc532238453)

# 第1章 概述

## 1.1 Hello简介

P2P：

Program：编写代码保存成hello.c。

Process：将hello.c预处理、编译、汇编、链接成为可执目标程序 hello。

020：

shell执行execve创建hello进程，映射虚拟内存。进入程序入口执行目标代码，CPU为程序分配时间片，高速缓存机制加速程序的运行。当程序运行结束后，shell回收hello进程，内核删除相关数据结构。

## 1.2 环境与工具

硬件环境：

Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz

16GB RAM

1TB HDD + 512G SSD

软件环境：

Windows 10 21H1

Ubuntu 20.04 LTS

开发工具：

VSCode，CodeBlocks，gcc，gdb，readelf

## 1.3 中间结果

列出你为编写本论文，生成的中间结果文件的名字，文件的作用等。

## 1.4 本章小结

**（第1章0.5分）**

# 第2章 预处理

## 2.1 预处理的概念与作用

概念：

预处理是C语言的一个重要功能，它由预处理程序负责完成。当对一个源文件进行编译时，系统将自动引用预处理程序对源程序中的预处理部分作处理，处理完毕自动进入对源程序的编译。 C语言提供多种预处理功能，主要处理#开始的预编译指令，如宏定义(#define)、文件包含(#include)、条件编译(#ifdef)等。

作用：

* 条件编译：条件编译的功能是根据条件有选择性的保留或者放弃源文件中的内容。常见的条件包含#if、#ifdef、#ifndef指令开始，以#endif结束。用#undef 指令可对用#define定义的标识符取消定义。
* 源文件包含：源文件包含指令的功能是搜索指定的文件，并将它的内容包含进来，放在当前所在的位置。源文件包含有两种，包含系统文件以及用户自定义文件。
* 宏替换：宏的作用是把一个标识符指定为其他一些成为替换列表的预处理记号，当这个标识符出现在后面的文本中时，将用对应的预处理记号把它替换掉，宏的本质是替换。
* 行控制：行控制指令以"#"和“line”引导，后面是行号和可选的字面串。它用于改变预定义宏"\_\_LINE\_\_"的值，如果后面的字面串存在，则改变“\_\_FILE\_\_”的值。
* 抛错：抛错指令是以“#”和“error”引导，抛错指令用于在预处理期间发出一个诊断信息，在停止转换。抛错是人为的动作。
* 杂注：杂注指令用于向C实现传递额外的信息（编译选项），对程序的某些方面进行控制。
* 空指令：空指令只有一个“#”，自成一行，空指令的使用没有效果。

## 2.2在Ubuntu下预处理的命令

cpp hello.c > hello.i

## 2.3 Hello的预处理结果解析

（*以下格式自行编排，编辑时删除*）

## 2.4 本章小结

（*以下格式自行编排，编辑时删除*）

**（第2章0.5分）**

# 第3章 编译

## 3.1 编译的概念与作用

（*以下格式自行编排，编辑时删除*）

注意：这儿的编译是指从 .i 到 .s 即预处理后的文件到生成汇编语言程序

## 3.2 在Ubuntu下编译的命令

（*以下格式自行编排，编辑时删除*）

应截图，展示编译过程！

## 3.3 Hello的编译结果解析

（*以下格式自行编排，编辑时删除*）

此部分是重点，说明编译器是怎么处理C语言的各个数据类型以及各类操作的。应分3.3.1~ 3.3.x等按照类型和操作进行分析，**只要hello.s中出现的属于大作业PPT中P4给出的参考C数据与操作，都应解析**。

## 3.4 本章小结

（*以下格式自行编排，编辑时删除*）

**（第3章2分）**

# 第4章 汇编

## 4.1 汇编的概念与作用

（*以下格式自行编排，编辑时删除*）

注意：这儿的汇编是指从 .s 到 .o 即编译后的文件到生成机器语言二进制程序的过程。

## 4.2 在Ubuntu下汇编的命令

（*以下格式自行编排，编辑时删除*）

应截图，展示汇编过程！

## 4.3 可重定位目标elf格式

分析hello.o的ELF格式，用readelf等列出其各节的基本信息，特别是重定位项目分析。

## 4.4 Hello.o的结果解析

（*以下格式自行编排，编辑时删除*）

objdump -d -r hello.o 分析hello.o的反汇编，并请与第3章的 hello.s进行对照分析。

说明机器语言的构成，与汇编语言的映射关系。特别是机器语言中的操作数与汇编语言不一致，特别是分支转移函数调用等。

## 4.5 本章小结

（*以下格式自行编排，编辑时删除*）

**（第4章1分）**

# 第5章 链接

## 5.1 链接的概念与作用

（*以下格式自行编排，编辑时删除*）

注意：这儿的链接是指从 hello.o 到hello生成过程。

## 5.2 在Ubuntu下链接的命令

（*以下格式自行编排，编辑时删除*）

使用ld的链接命令，应截图，展示汇编过程！ 注意不只连接hello.o文件

## 5.3 可执行目标文件hello的格式

分析hello的ELF格式，用readelf等列出其各段的基本信息，包括各段的起始地址，大小等信息。

## 5.4 hello的虚拟地址空间

使用edb加载hello，查看本进程的虚拟地址空间各段信息，并与5.3对照分析说明。

## 5.5 链接的重定位过程分析

（*以下格式自行编排，编辑时删除*）

objdump -d -r hello 分析hello与hello.o的不同，说明链接的过程。

结合hello.o的重定位项目，分析hello中对其怎么重定位的。

## 5.6 hello的执行流程

（*以下格式自行编排，编辑时删除*）

使用edb执行hello，说明从加载hello到\_start，到call main,以及程序终止的所有过程。请列出其调用与跳转的各个子程序名或程序地址。

## 5.7 Hello的动态链接分析

（*以下格式自行编排，编辑时删除*）

分析hello程序的动态链接项目，通过edb调试，分析在dl\_init前后，这些项目的内容变化。要截图标识说明。

## 5.8 本章小结

（*以下格式自行编排，编辑时删除*）

**（第5章1分）**

# 第6章 hello进程管理

## 6.1 进程的概念与作用

（*以下格式自行编排，编辑时删除*）

## 6.2 简述壳Shell-bash的作用与处理流程

（*以下格式自行编排，编辑时删除*）

## 6.3 Hello的fork进程创建过程

（*以下格式自行编排，编辑时删除*）

## 6.4 Hello的execve过程

（*以下格式自行编排，编辑时删除*）

## 6.5 Hello的进程执行

（*以下格式自行编排，编辑时删除*）

结合进程上下文信息、进程时间片，阐述进程调度的过程，用户态与核心态转换等等。

## 6.6 hello的异常与信号处理

（*以下格式自行编排，编辑时删除*）

hello执行过程中会出现哪几类异常，会产生哪些信号，又怎么处理的。

程序运行过程中可以按键盘，如不停乱按，包括回车，Ctrl-Z，Ctrl-C等，Ctrl-z后可以运行ps jobs pstree fg kill 等命令，请分别给出各命令及运行结截屏，说明异常与信号的处理。

## 6.7本章小结

（*以下格式自行编排，编辑时删除*）

**（第6章1分）**

# 第7章 hello的存储管理

## 7.1 hello的存储器地址空间

（*以下格式自行编排，编辑时删除*）

结合hello说明逻辑地址、线性地址、虚拟地址、物理地址的概念。

## 7.2 Intel逻辑地址到线性地址的变换-段式管理

（*以下格式自行编排，编辑时删除*）

## 7.3 Hello的线性地址到物理地址的变换-页式管理

（*以下格式自行编排，编辑时删除*）

## 7.4 TLB与四级页表支持下的VA到PA的变换

（*以下格式自行编排，编辑时删除*）

## 7.5 三级Cache支持下的物理内存访问

（*以下格式自行编排，编辑时删除*）

## 7.6 hello进程fork时的内存映射

（*以下格式自行编排，编辑时删除*）

## 7.7 hello进程execve时的内存映射

（*以下格式自行编排，编辑时删除*）

## 7.8 缺页故障与缺页中断处理

（*以下格式自行编排，编辑时删除*）

## 7.9动态存储分配管理

（*以下格式自行编排，编辑时删除*）

*Printf会调用malloc，请简述动态内存管理的基本方法与策略。*

## 7.10本章小结

（*以下格式自行编排，编辑时删除*）

**（第7章 2分）**

# 第8章 hello的IO管理

## 8.1 Linux的IO设备管理方法

（*以下格式自行编排，编辑时删除*）

设备的模型化：文件

设备管理：unix io接口

## 8.2 简述Unix IO接口及其函数

（*以下格式自行编排，编辑时删除*）

## 8.3 printf的实现分析

（*以下格式自行编排，编辑时删除*）

<https://www.cnblogs.com/pianist/p/3315801.html>

从vsprintf生成显示信息，到write系统函数，到陷阱-系统调用 int 0x80或syscall.

字符显示驱动子程序：从ASCII到字模库到显示vram（存储每一个点的RGB颜色信息）。

显示芯片按照刷新频率逐行读取vram，并通过信号线向液晶显示器传输每一个点（RGB分量）。

## 8.4 getchar的实现分析

（*以下格式自行编排，编辑时删除*）

异步异常-键盘中断的处理：键盘中断处理子程序。接受按键扫描码转成ascii码，保存到系统的键盘缓冲区。

getchar等调用read系统函数，通过系统调用读取按键ascii码，直到接受到回车键才返回。

## 8.5本章小结

（*以下格式自行编排，编辑时删除*）

**（第8章1分）**

# 结论

用计算机系统的语言，逐条总结hello所经历的过程。

你对计算机系统的设计与实现的深切感悟，你的创新理念，如新的设计与实现方法。

**（结论0分，缺失 -1分，根据内容酌情加分）**

# 附件

列出所有的中间产物的文件名，并予以说明起作用。

**（附件0分，缺失 -1分）**

# 参考文献

**为完成本次大作业你翻阅的书籍与网站等**

[1] 林来兴. 空间控制技术[M]. 北京：中国宇航出版社，1992：25-42.

[2] 辛希孟. 信息技术与信息服务国际研讨会论文集：A集[C]. 北京：中国科学出版社，1999.

[3] 赵耀东. 新时代的工业工程师[M/OL]. 台北：天下文化出版社，1998 [1998-09-26]. http://www.ie.nthu.edu.tw/info/ie.newie.htm（Big5）.

[4] 谌颖. 空间交会控制理论与方法研究[D]. 哈尔滨：哈尔滨工业大学，1992：8-13.

[5] KANAMORI H. Shaking Without Quaking[J]. Science，1998，279（5359）：2063-2064.

[6] CHRISTINE M. Plant Physiology: Plant Biology in the Genome Era[J/OL]. Science，1998，281：331-332[1998-09-23]. http://www.sciencemag.org/cgi/ collection/anatmorp.

**（参考文献0分，缺失 -1分）**