广东工业大学

计算机科学与技术学院

《Linux技术》

期末作业



|  |  |
| --- | --- |
| 课程名称： | Linux技术 |
| 学生姓名： | 王宁 |
| 班级学号： | 4-3120005084 |
| 填写日期： | 2023-05-23 |
| 成绩： |  |

**Linux教程 期末作业**

**（第一题50分，第二题30分，第三题20分）**

目录

[1 第一题 2](#_Toc135751274)

[1.1 实验题目 2](#_Toc135751275)

[1.2 实验报告 2](#_Toc135751276)

[1.2.1 设计思路 2](#_Toc135751277)

[1.2.2 文件依赖关系 3](#_Toc135751278)

[1.2.3 编译和执行结果 4](#_Toc135751279)

[1.2.4 GDB入门操作 6](#_Toc135751280)

[1.2.5 附录 7](#_Toc135751281)

[2 第二题 8](#_Toc135751282)

[2.1 实验题目 8](#_Toc135751283)

[2.2 实验报告 8](#_Toc135751284)

[2.2.1 设计思路 8](#_Toc135751285)

[2.2.2 运行结果 9](#_Toc135751286)

[2.2.3 源代码 11](#_Toc135751287)

[3 第三题 12](#_Toc135751288)

[3.1 实验题目 12](#_Toc135751289)

[3.2 实验报告 12](#_Toc135751290)

[3.2.1 使用vi修改m1.c 12](#_Toc135751291)

[3.2.2 错误汇总 14](#_Toc135751292)

[3.2.3 修改前的m1.c 14](#_Toc135751293)

[3.2.4 修改完成的m1.c 15](#_Toc135751294)

[3.2.5 编译和执行结果 16](#_Toc135751295)

# 第一题

## 1.1 实验题目

**用c语言设计命令行交互菜单，根据用户输入实现相应功能。菜单如下：**

------------------

C 交互菜单

------------------

A:查看CPU硬件信息

B:查看进程运行信息

C:查看内存硬件信息、内存使用信息

D:查看存储设备硬件信息、存储设备使用信息

E:退出

------------------

具体要求如下：

(1)将用户输入选项与显示信息写入文件log.txt

(2)使用makefile管理项目（写在多个.c .h源文件中）

(3)使用gcc、gdb进行编译和调试

(4)录制视频展示运行过程和部分调试过程

(5)报告应包含：文件依赖关系图、设计思路、编译和执行结果

## 1.2 实验报告

### 1.2.1 设计思路

(1) main函数使用死循环运行程序，接收选项参数，根据相应的选项参数实现题目要求的功能。

(2) 根据题目要求的功能分别编写对应的C语言文件函数实现，在main函数中根据选项调用对应的C语言文件函数。

(3) 其中的C语言文件实现功能：

|  |  |
| --- | --- |
| **文件** | **功能** |
| menu.h、menu.c | 实现菜单显示、日志输出相关功能 |
| cpu.h、cpu.c | 实现处理CPU信息的函数 |
| process.h、process.c | 实现处理进程信息的函数 |
| memory.h、memory.c | 实现处理内存信息的函数 |
| storage.h、storage.c | 实现处理存储设备的函数 |
| exit.h、exit.c | 实现退出程序功能 |
| main.c | 实现主程序逻辑 |
| makefile | 自动化构建程序 |

(4) 实现题目要求功能对应的Linux命令或C库函数：

|  |  |
| --- | --- |
| **功能** | **Linux命令或C库函数** |
| 查看CPU硬件信息 | lscpu |
| 查看进程运行信息 | ps aux |
| 查看内存硬件信息、内存使用信息 | free -h |
| 查看存储设备硬件信息、存储设备使用信息 | df -h |
| 退出 | exit(0) |

(5) 编写makefile文件自动编译构建项目。

### 1.2.2 文件依赖关系

(1) 文件依赖关系图：



(2) makefile文件：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

### 1.2.3 编译和执行结果

(1) 编译

cd makefile所在目录

# 编译 make / make all

make

文本

描述已自动生成

(2) 执行结果

# 授予执行权限

chmod a+x program\_1

# 运行可执行文件

./program\_1

文本

描述已自动生成

其中A、B、C、D选项执行对应功能，E选项退出程序，其余选项均无效，给出提示。所有选项命令及相应输出信息都会被写入到log.txt文件。

图形用户界面, 文本, 应用程序

描述已自动生成

文本

中度可信度描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

背景图案

低可信度描述已自动生成

(3) 清理

# 运行Makefile中的clean命令：删除所有目标文件、可执行文件，询问删除log.txt

make clean

文本

中度可信度描述已自动生成

### 1.2.4 GDB入门操作

GDB入门操作命令：

|  |  |  |
| --- | --- | --- |
| 命令 | 简写形式 | 说明 |
| backtrace | bt、where | 显示backtrace |
| break | b | 设置断点 |
| continue | c、count | 继续执行 |
| delete | d | 删除断点 |
| finish |  | 运行到函数结束 |
| info breakpoints |  | 显示断点信息 |
| next | n | 执行下一行 |
| print | p | 显示表达式 |
| run | r | 运行程序 |
| step | s | 一次执行一行，包括函数内部 |
| x |  | 显示内存内容 |
| until | u | 执行到指定行 |
| 其他命令 |  |  |
| directory | dir | 插入目录 |
| disable | dis | 禁用断点 |
| down | do | 在当前调用的栈帧中选择要显示的栈帧 |
| edit | e | 编辑文件或者函数 |
| from | f | 选择要显示的栈帧 |
| forward-search | fo | 向前搜索 |
| generate-core-file | gcore | 生成内核转存储 |
| help | h | 显示帮助一览 |
| info | i | 显示信息 |
| list | l | 显示函数或行 |
| nexti | ni | 执行下一行（以汇编代码为单位） |
| print-object | po | 显示目标信息 |
| sharelibrary | share | 加载共享的符号 |
| stepi | si | 执行下一行 |

### 1.2.5 附录

代码：

图形用户界面, 文本

描述已自动生成

运行过程和部分调试过程视频：

图形用户界面, 文本, 应用程序

描述已自动生成

# 第二题

## 2.1 实验题目

**设计shell脚本程序，运行结果如下：**

------------------

Linux 23/06/11

------------------

A:查看进程信息

B:查看用户信息

C:显示当前目录信息

D:显示计算机名、Linux内核版本、当前IP地址

E:退出

------------------

(1)第一行显示系统日期：年/月/日

(2)当用户输入相应数字执行相应的功能

(3)报告应包含：设计思路、源代码（含注释）、运行结果

## 2.2 实验报告

### 2.2.1 设计思路

(1) 程序死循环执行，当给出退出命令时exit 0结束程序。

(2) 系统日期23/06/11实现：

获取年 | 截取后2位数字，获取其他日期拼接成DATE。

图片包含 表格

描述已自动生成

(3) 显示菜单列表：

echo -e识别转义字符\t

文本

描述已自动生成

(4) 读取用户输入，根据输入选项执行相应功能：

|  |  |
| --- | --- |
| **用户输入** | **功能** |
| A、a、1 | 查看进程信息 |
| B、b、2 | 查看用户信息 |
| C、c、3 | 显示当前目录信息 |
| D、d、4 | 显示计算机名、Linux内核版本、当前IP地址 |
| E、e、5 | 退出 |
| \*（其余） | （无效选项） |

(5) 实现功能所需Linux命令：

|  |  |
| --- | --- |
| **功能** | **Linux命令** |
| 查看进程信息 | ps aux |
| 查看用户信息 | who |
| 显示当前目录信息 | ls -l |
| 显示计算机名 | hostname |
| 显示Linux内核版本 | uname -r |
| 显示当前IP地址 | ip addr show | grep 'inet ' | awk '{print $2}' |
| 退出 | exit 0 |
| （无效选项） | （无效选项） |

显示当前IP地址：



### 2.2.2 运行结果

# 进入startup.sh所在目录，授予权限并执行

chmod a+x startup.sh && ./startup.sh

文本

描述已自动生成

(1) 查看进程信息：

表格

描述已自动生成

(2) 查看用户信息：

文本, 信件

描述已自动生成

(3) 显示当前目录信息：

文本

描述已自动生成

(4) 显示计算机名、Linux内核版本、当前IP地址

文本

中度可信度描述已自动生成

(5) 无效选项：

文本, 信件

描述已自动生成

(6) 退出：

文本

描述已自动生成

### 2.2.3 源代码

文本

描述已自动生成

# 第三题

## 3.1 实验题目

**vim命令**

要求：

(1)使用vi修改存在大量错误的c语言文件 m1.c，使其能够正确编译和执行

(2)调整代码缩进，从而提高可读性

(3)按类别指出m1.c存在的错误，并用vi改正

(4)报告应包含：错误汇总、针对每一种错误所执行的vi操作、修改完成的源代码、编译和执行结果。

提示：m1.c的错误可能包括变量拼写错误、不正确的分号、括号位置不正确等。

## 3.2 实验报告

### 3.2.1 使用vi修改m1.c

(1) 第2、3行头文件拼写错误

// vim打开m1.c

vim m1.c

// 显示行号

:set nu

// 1.头文件拼写错误

i

// 进入插入模式，修改第2、3行的头文件

(2) 修改中文标点符号

// 2.修改中文标点符号

// 按ESC进入命令模式

// 全文替换’；’、’，’为’;’、’,’

:%s/；/;/g

:%s/，/,/g

(3) 修改占位符”% s”为”%s”

// 3.修改占位符”% s”为”%s

16gg f % i

25gg f % i

(4) 变量名nbytes的拼写错误

// 4.变量名nbytes的拼写错误

// 跳转至第9行，替换当前行”nbyte3”为” nbytes”

9gg

:s/nbyte3/ nbytes

(5) 第15、16行argv[l]数组索引错误

// 5.第15、16行argv[l]数组索引错误

// 模式匹配

/argv[l

// 按i进入插入模式，修改argv[l]为argv[1]

i

// 按ESC进入命令模式，按n继续下一个匹配

ESC n

// 按i进入插入模式，修改argv[l]为argv[1]

i

(6) 第28行for循环{}不成对

// 6. 第28行for循环{}不成对

// 按ESC进入命令模式，跳转至第30行

ESC 30gg

// 按i进入插入模式，删除}

i

(7) 修改第12行)位置错误

// 7.修改第12行)位置错误

// 按ESC进入命令模式

ESC

// 跳转至12行行尾，修改)位置

12gg $ i

(8) 调整代码缩进

// 8.调整代码缩进

// 按ESC进入命令模式

ESC

// 指定文件类型为C语言

:set ft=c

// 选择整个文件内容

ggVG

// 按下=a，进行自动缩进和格式化

=a

// 保存并推出

:wq

### 3.2.2 错误汇总

调试完毕后，发现错误主要集中于：

(1) 修复了头文件的拼写错误。

(2) 移除了多余的空格和中文标点符号。

(3) 修正了变量名的拼写错误。

(4) 修正了错误消息的格式化。

(5) 调整代码缩进。

### 3.2.3 修改前的m1.c

文本

描述已自动生成

### 3.2.4 修改完成的m1.c

文本

低可信度描述已自动生成

### 3.2.5 编译和执行结果

M1.c的作用是将一个文件的内容复制到另一个文件，并在复制完成后显示当前目录的路径内容。

准备m1\_test.txt、m2\_dest.txt文件，其中m1\_test.txt内容为修改前的m1.c源文件，m2\_dest.txt为空文件。

**编译：**

// gcc编译

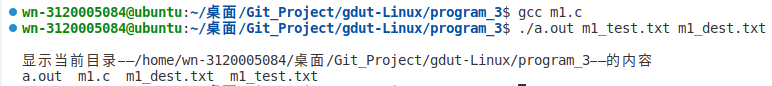
gcc m1.c

// 运行可执行文件，将m1\_test.txt复制到m1\_dest.txt

./a.out m1\_test.txt m1\_dest.txt

**执行结果：**

复制完成后显示当前目录的路径内容：



将m1\_test.txt复制到m1\_dest.txt：图形用户界面, 文本, 应用程序

描述已自动生成