

《嵌入式系统》实验报告 1

学号：21013134 姓名：徐昊博 班级：计 213 日期：2024.4.26

成绩：_____

指导教师：罗飞_____

实验名称：嵌入式 Linux 基础实验	实验地点：
实验仪器：ECS 云实验环境--Linux 实验环境	
<p>一、实验目的：</p> <ol style="list-style-type: none">1、掌握 Linux 各类命令的使用方法；2、熟悉 Linux 操作环境3、了解 Makefile 的基本概念和基本结构4、初步掌握编写简单 Makefile 的方法5、了解递归 Make 的编译过程6、初步掌握利用 GNU Make 编译应用程序的方法	
<p>二、实验内容：</p> <ol style="list-style-type: none">1. 基于实验指导手册，练习使用 Linux 常用命令，简述主要命令的含义 <p>ls: 列出目录中的文件和子目录。</p> <p>mkdir: 创建一个新目录。</p> <p>cd: 更改当前工作目录</p> <p>vi: 启动 vi 文本编辑器。</p> <p>cp: 复制文件或目录。</p> <p>less: 逐页查看文件内容。</p> <p>grep: 在文件中搜索匹配的文本模式。</p> <p>tar: 打包和压缩文件。</p> <p>rm: 删除文件或目录。</p> <p>date: 显示当前日期和时间。</p> <p>free: 显示系统内存的使用情况。</p> <p>df: 显示磁盘空间的使用情况。</p> <p>完成实验指导手册中的内容二--四，进而完成如下工作：</p> <ol style="list-style-type: none">2. 新建项目目录 project-1，完成如下功能： <ol style="list-style-type: none">1) 在该项目中编写 hello.c 程序，该程序输出 “Hello,world! ”；2) 通过 gcc 命令编译该文件，运行该可执行程序。 <p>简述实验步骤。</p> <p>(1) 新建一个工程目录，并进入到该目录</p> <p>ecust@Ubuntu mkdir project-1</p> <p>ecust@Ubuntu cd project-1</p> <p>(2) 利用文本编辑器创建 hello.c 文件</p> <p>ecust@Ubuntu ~/project-1\$ vim hello.c</p> <p>然后按一下 “i” 键，随后输入以下内容</p> <p>#include <stdio.h></p>	

```
int main(){  
    printf("Hello,World!\n");  
    return 0;  
}
```

编辑完成后，按“Esc”键，退出输入模式；

接着输入“: wq”，表示保存以上文件后退出。

(3) 通过 gcc 命令手动编译 hello 应用程序

```
ecust@Ubuntu~/project-1$ gcc -c hello.c
```

```
ecust@Ubuntu~/project-1$ gcc hello.o -o hello
```

运行 hello 程序：

```
ecust@Ubuntu~/project-1$ ./hello
```

3. 新建项目目录 project-2，完成如下功能：

1) 在该项目中编写 makefile-hello.c 程序；

2) 该程序输出“Hello,world for Makefile! ”；

3) 编写 Makefile，通过 Makefile 编译 makefile-hello.c，并运行可执行程序。

简述实验步骤。

(1) 新建一个工程目录，并进入到该目录

```
ecust@Ubuntu mkdir project-2
```

```
ecust@Ubuntu cd project-2
```

(2) 利用文本编辑器创建 hello.c 文件

```
ecust@Ubuntu~/project-2$ vim hello.c
```

然后按一下“i”键，随后输入以下内容

```
#include <stdio.h>
```

```
int main(){  
    printf( "Hello,world for Makefile!\n" );  
    return 0;
```

```
}
```

(3) 利用文本编辑器按照 2 的步骤分别创建一个 makefile 文件，并将其保存到与 makefile-hello.c 相同的目录下。

```
CC=gcc
```

```
CFLAGS=
```

```
all: makefile-hello
```

```
makefile-hello: makefile-hello.o
```

```
$(CC) $(CFLAGS) makefile-hello.o -o makefile-hello
```

```
makefile-hello.o: makefile-hello.c
```

```
$(CC) $(CFLAGS) -c makefile-hello.c -o makefile-hello.o
```

```
clean:
```

```
rm -rf makefile-hello*.o
```

(4) 执行指令：

```
ecust@Ubuntu~/project-2$ make
```

```
ecust@Ubuntu~/project-2$ ls
```

```
ecust@Ubuntu~/project-2$ ./makefile-hello
```

4. 新建项目目录 project-3, 完成如下功能:

- 1) 编写 C 语言文件 file1.c、file2.c 和相关的头文件;
- 2) file1.c 输出: “This is the first file!”;
- 3) file2.c 输出: “This is the second file!”;
- 4) 编写 Makefile, 编译该工程, 并运行可执行程序, 依次输出:

This is the first file!

This is the second file!

简述实验步骤。

(1) 新建一个工程目录

ecust@Ubuntu mkdir project-3

(2) 按照之前题目的步骤分别创建三个文件 file1.c、file2.c、file2.h 和 makefile

file1.c 的内容如下:

```
#include <stdio.h>
```

```
int main(){
```

```
    printf( “ This is the first file!\n” );
```

```
    test2();
```

```
    return 0;
```

```
}
```

file2.c 的内容如下:

```
#include <stdio.h>
```

```
#include “file2.h”
```

```
void test2(void){
```

```
    printf( “ This is the second file!” );
```

```
}
```

file2.h 的内容如下:

```
void test2(void);
```

makefile 的内容如下:

```
CC=gcc
```

```
CFLAGS=
```

```
OBJS=file1.o file2.o
```

```
all: file
```

```
file: $(OBJS)
```

```
    $(CC) $(CFLAGS) $^ -o $@
```

```
file1.o: file1.c
```

```
    $(CC) $(CFLAGS) -c $< -o $@
```

```
file2.o: file2.c
```

```
    $(CC) $(CFLAGS) -c $< -o $@
```

```
clean:
```

```
    rm -rf file*.o
```

(3) 先后执行如下命令

ecust@Ubuntu~/project-3\$ make

ecust@Ubuntu~/project-3\$ ls

ecust@Ubuntu~/project-3\$./file

5. Linux 命令综合：根据各人学号 or 姓名，为自己新建一个工作目录，并利用 pwd 命令查看自己的工作目录；将前述工程 project-1 到 project-3 打包成 back.tar.gz；简述实验步骤：

（1）新建工作目录：使用 mkdir 命令创建一个新的工作目录，指令为：

`mkdir xuhaobo-21013134`

（2）查看工作目录：使用 pwd 命令查看当前工作目录的绝对路径，指令为：

`pwd`

（3）使用 tar 命令将 project-1 到 project-3 打包成一个压缩文件。首先进入包含这三个工程目录的父目录，然后运行以下命令即可：

`tar -zcvf back.tar.gz project-1 project-2 project-3`