



- 课上测试
  - ch01
  - ch02
    - 作业题目：Linux命令实践（AI）
      - 学习Linux核心命令（10 分）
      - 作业要求（9'）
      - 作业提交要求（1'）
    - 作业题目：C编程工具测试（AI）
      - 完成下面任务（29 分）
      - 作业提交要求（1'）
    - 作业题目: gdb调试
      - 完成下面任务（9 分）
      - 作业提交要求（1'）
    - 作业题目: 位运算
      - 完成下面任务（29 分）
      - 作业提交要求（1'）
  - ch03
  - ch04
  - ch05
  - ch06

## 课上测试

---

### ch01

---

### ch02

---

#### 作业题目：Linux命令实践（AI）

##### 学习Linux核心命令（10 分）

- ls
- who
- pwd
- cd
- man
- whereis
- which
- find
- locate
- grep

## 作业要求 (9')

- 参考课件，针对上面核心命令重复老师的演示内容 (3')
- 选择自己习惯的 AI 工具，针对上面核心命令，让 AI 给自己生成至少5个例子，然后自己进行验证，总结 (3')
- 让 AI工具推荐一些重要命令，然后选择自己认为会高频应用的至少 3个命令（重点推荐 echo, od, sort, bc, ls）进行自主深入学习，每个命令不少于 5 个例子，提交验证，实践过程和总结 (3')

## 作业提交要求 (1')

0. 记录实践过程和 AI 问答过程，尽量不要截图，给出文本内容
1. (选做)推荐所有作业托管到 [gitee](#)或 [github](#) 上
2. (必做)提交作业 markdown文档，命名为“学号-姓名-作业题目.md”
3. (必做)提交作业 markdown文档转成的 PDF 文件，命名为“学号-姓名-作业题目.pdf”

## 作业题目：C编程工具测试 (AI)

### 完成下面任务 (29 分)

0. 在 Ubuntu 活 openEuler 中完成任务（推荐openEuler）
1. 编写C代码hello.c，打印“hello 你的 8 位学号”,给出编译预处理，编译，汇编，链接的gcc命令，及其运行结果，使用 git 记录过程。（5 分）
2. <https://gitee.com/>下载课程代码，编译运行老师课上给出大端小端的代码，给出运行结果。(5分)

```
$ pwd
bestidiocs2024/ch02/big-little-endian
$ ls
cebitwise.c  cepointer.c
cebitfields.c  celibrary.c  ceunion.c
```

3. 新建目录，拷贝要求2中代码，把代码中的 main 函数都去掉，编写ble.h(包含五个判断函数), main.c 通过命令行参数调用，比如 ./main 1 调用cebitwise.c 中的功能。编译运行代码，给出运行结果。使用 git 记录过程 (5分)
4. 新建目录，拷贝要求3中代码，制作静态库 libble.a,并调用 libble.a,编译运行代码，给出运行结果。使用 git 记录过程 (5分)
5. 新建目录，拷贝要求3中代码，制作静态库 libble.a,并调用 libble.a,编译运行代码，给出运行结果。使用 git 记录过程 (5分)
6. 新建目录，按下面目录拷贝要求3中代码，编写 Makefile 编译运行代码，给出运行结果。使用 git 记录过程 (4分)

```
tree
.
├── Makefile
├── bin: 可执行文件
├── docs: 文档
├── include: 头文件
├── lib: 动态库，静态库
└── src: 源代码
```

7. 提交 git log 结果。

## 作业提交要求 (1')

- 0. 记录实践过程和 AI 问答过程，尽量不要截图，给出文本内容
- 1. (选做)推荐所有作业托管到 [gitee](#)或 [github](#) 上
- 2. (必做)提交作业 markdown文档，命名为“[学号-姓名-作业题目.md](#)”
- 3. (必做)提交作业 markdown文档转成的 PDF 文件，命名为“[学号-姓名-作业题目.pdf](#)”

## 作业题目: gdb调试

### 完成下面任务 (9 分)

- 0. 在 Ubuntu 活 openEuler 中完成任务（推荐openEuler）
- 1. <https://gitee.com/下载课程代码>，编译运行 "bestidiocs2024/ch02/c/testgdb.c"，记录编译运行结果（2分）
- 2. 参考<https://www.cnblogs.com/rocedu/p/16714289.html>调试代码，至少涵盖 4 种断点（行断点，函数断点，临时断点，条件断点）（5分）
- 3. 使用单步运行等其他命令。（2分）

## 作业提交要求 (1')

- 0. 记录实践过程和 AI 问答过程，尽量不要截图，给出文本内容
- 1. (选做)推荐所有作业托管到 [gitee](#)或 [github](#) 上
- 2. (必做)提交作业 markdown文档，命名为“[学号-姓名-作业题目.md](#)”
- 3. (必做)提交作业 markdown文档转成的 PDF 文件，命名为“[学号-姓名-作业题目.pdf](#)”

## 作业题目: 位运算

### 完成下面任务 (29 分)

密码系统有时间戳的需求，因此密码系统有实时钟芯片。假设实时钟芯片全局变量unsigned short TIME定义。

0. 在 Ubuntu 活 openEuler 中完成任务（推荐openEuler）

1. 16 位TIME 的含义

```

      TIME
(bit15)  ++++++ ++++++ ++++++(bit0)
          |||||  |||||  |||||
          |||||  |||||  +----- second/2
          |||||  +----- minute
          +----- hour
```

- 2. 使用位运算编写并调用下面函数，把当前时间(使用 C库函数获得)设置到 TIME 中，给出代码，使用 git 记录过程。（12 分）

```
void setHour(int h)
void setMinute(int m)
void setSecond(int s)
```

3. 使用位运算编写并调用下面函数，读取TIME 中的时间，给出代码，使用 git 记录过程。（12 分）

```
void setHour(int h)
void setMinute(int m)
void setSecond(int s)
```

4. 完成main.c测试模块，测试你完成的代码，给出代码，编译运行结果，使用 git 记录过程。（5'）

- 获取系统时间设置TIME
- 读取你设置的结果
- 设置minute 为你学号最后两位
- 读取minute

## 作业提交要求 (1')

0. 记录实践过程和 AI 问答过程，尽量不要截图，给出文本内容
1. (选做)推荐所有作业托管到 [gitee](#)或 [github](#) 上
2. (必做)提交作业 markdown文档，命名为“学号-姓名-作业题目.md”
3. (必做)提交作业 markdown文档转成的 PDF 文件，命名为“学号-姓名-作业题目.pdf”

## ch03

---

## ch04

---

## ch05

---

## ch06

---

