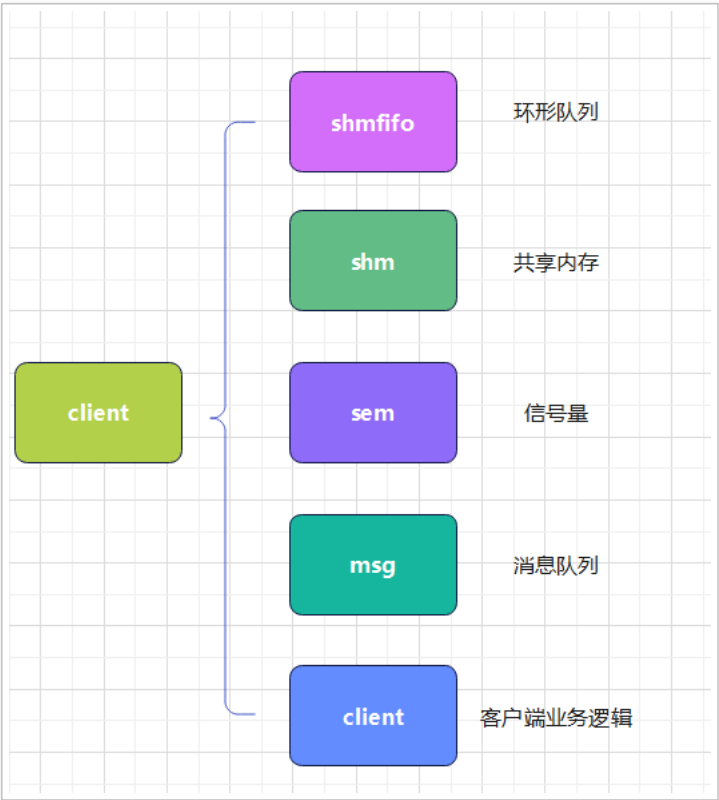


4.1 项目模块整合与数据结构设计_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 4.1 项目模块整合与数据结构设计涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。





- 创建两个目录，客户端与服务端，并将相应的模块到对应的目录中

- client : 客户端目录 (msg 模块在后面实现), 需要添加 client.h 与 client.c

```
ben@ubuntu:~/class/week13/project_part/version1/client$ ls
client.c client.h main.c Makefile sem.c sem.h shm.c shmfifo.c shmfifo.h shm.h
ben@ubuntu:~/class/week13/project_part/version1/client$
```

- server : 服务器目录 (msg 模块与 hashmap 模块在后面实现), 需要添加 server.h 与 server.c

```
ben@ubuntu:~/class/week13/project_part/version1/server$ ls
main.c sem.c server.c shm.c shmfifo.h thread_pool.c
Makefile sem.h server.h shmfifo.c shm.h thread_pool.h
ben@ubuntu:~/class/week13/project_part/version1/server$
```

- 创建两个临时文件，用于创建共享内存与消息队列
 - 创建的路径为 client 与 server 上一级目录中

```
ben@ubuntu:~/class/week13/workspace/project$ ls
client client_pub msg server shm
ben@ubuntu:~/class/week13/workspace/project$
```

- 编写 Makefile

- client Makefile

```
OBJS := sem.o shmfifo.o shm.o client.o main.o msg.o
```

```
client:$(OBJS)
@gcc $^ -o $@ -lpthread
@echo "Done."
```

```
%o : %.c
@gcc -c $< -o $@
```

```
clean:
@rm *.o client
@echo "Clean."
```

- server Makefile

```
OBJS := sem.o shmfifo.o shm.o server.o main.o thread_pool.o
```

```
server:$(OBJS)
@gcc $^ -o $@ -lpthread
@echo "Done."
```

```
%o : %.c
@gcc -c $< -o $@
clean:
@rm *.o server
@echo "Clean."
```

- 修改 shm.c 创建共享内存的文件名为 shm 临时文件, 客户端与服务器的 shm 模块要同步

```
#include "shm.h"

#define PATHNAME "../shm"
#define PRO_ID 101
```

```
enum work_mode {
    PUBLISH = 0,
    SUBSCRIBE,
};
```

- 传输的数据包数据结构设计
 - 主要用于将 主题、进程 id、工作模式、消息内容进行封装在传输

```
#define TOPIC_SZ 64
#define CONTENT_SZ 64

typedef struct packet{
    char topic[TOPIC_SZ];
    pid_t pid;
    enum work_mode mode;
    char content[CONTENT_SZ];
}packet_t;
```

- 具体在 client.h 与 server.h 添加相应的数据结构即可

全文完

本文由 简悦 SimpRead 优化, 用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta, 点击查看详细说明

