**打包解包v0.1**

**研发七部**

模块名：<weedong/core/packet/packet.h>

依赖: <weedong/core/os.h>

简介(详细请看<weedong/core/packet/packet.h> )：

打包和解包数据

结构(类)：

struct packet\_t; //时间结构

相关函数：

// 分配空间

packet\_t\* packet\_alloc(size\_t length);

// 加入新的包，返回-1表示失败，其他表示写入的数据长度

int packet\_join(packet\_t\* p, const void\* in\_data, size\_t in\_max\_size);

int packet\_join\_struct(packet\_t\* p, const void\* in\_data, size\_t in\_max\_size);

// 从包中获取数据，返回-1表示失败，其他表示写入的数据长度

int packet\_take(packet\_t\* p, void\* out\_data, size\_t out\_max\_size);

int packet\_take\_struct(packet\_t\* p, void\* out\_data, size\_t out\_max\_size);

// 回收packet

void packet\_dispose(packet\_t\* p);

// 直接打包成大的数据

packet\_t\* packet\_pack(const void\* in\_data, size\_t in\_max\_size);

示例：

#include <weedong/core/packet/packet.h>

main()

{

// 打包数据

int token = 0x0101;

short data = 100;

char text[] = “hello, world\0”;

packet\_t\* p = packet\_alloc(256); // 分配包

packet\_join(p, &token, sizeof(int)); // 写入int数据

packet\_join(p, &data, sizeof(short)); // 写入int数据

packet\_join\_struct(p, text, strlen(text)); // 写入字符串或者结构

// 测试输出

int out\_token = 0;

short out\_data = 0;

char out\_text[128] = “\0”;

packet\_take(p, &out\_token, sizeof(int)); //从包拿int数据

packet\_take(p, &out\_data, sizeof(short)); // 从包拿short数据

packet\_take\_struct(p, out\_text, sizeof(out\_text)); // 从包拿字符串或者结构

packet\_dispose(p); // 释放数据

}