

链

表

操

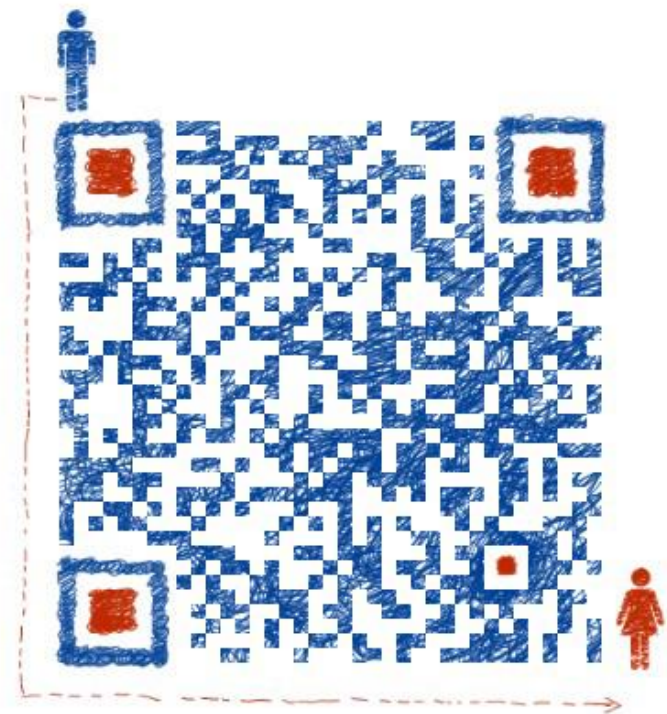
作

— Linux

无线传感器网项目实战



公众号:一口Linux



彭老师个人微信号

01

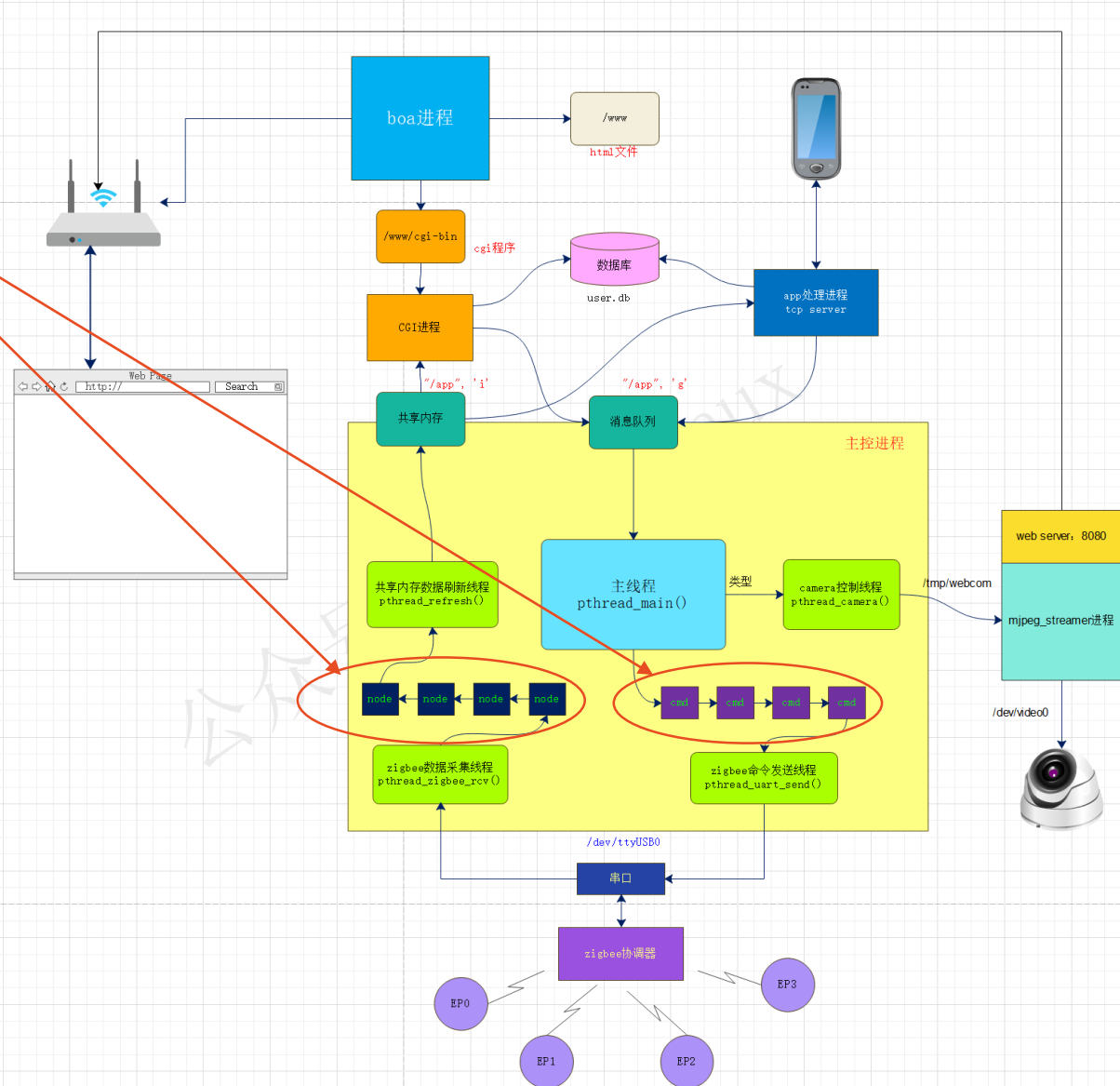
# 前言

# 链表重要性

- 在Linux内核里，大量使用了链表操作，list\_head
  - 设备管理、驱动管理、内存管理、等待队列、中断资源
- list\_head使用
  - 物联网课程备课专用\备课用\主控\0.内核链表list\_head操作.pdf

# 本项目链表

- 本项目中有2个地方使用到链表
- 均使用单链表
- 操作：
  - 插入节点，读取节点



关注公众号：—

02

# 链表操作

# 链表操作

- 1.创建链表

- linklist CreateEmptyLinklist()

- 2.插入节点

- int InsertLinknode (link\_datatype x)

- 3.提取一个节点

- linklist GetLinknode (linklist h)

```
typedef struct msg_pack
{
    char msg_type;
    char text[27];
}link_datatype;
```

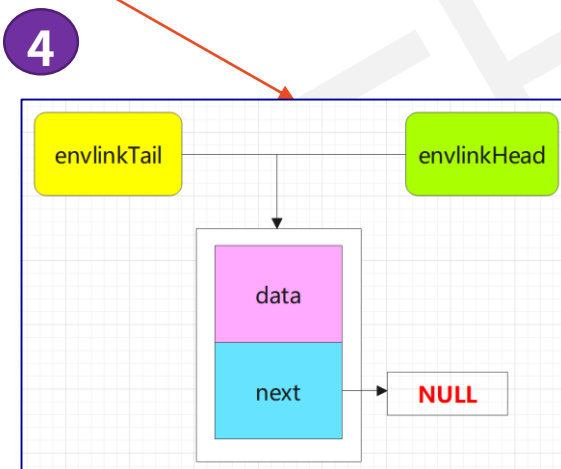
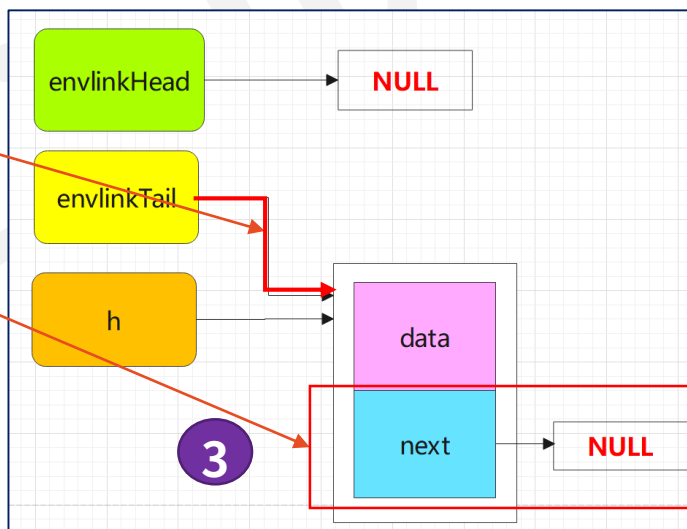
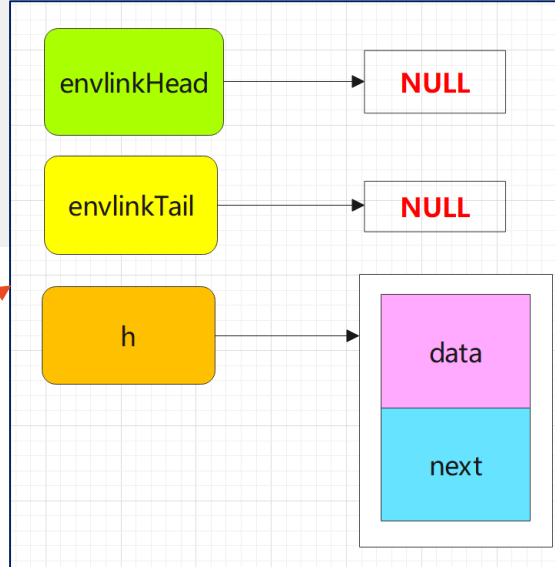
```
typedef struct _node_
{
    link_datatype data;
    struct _node_ *next;
}linknode, *linklist;
```

# 1.创建链表

```
linklist envlinkHead, envlinkTail;
```

```
linklist CreateEmptyLinklist()  
{  
    linklist h;  
    h = (linklist)malloc (sizeof (linknode));  
    envlinkTail = h;  
    h->next = NULL;  
    return h;  
}
```

```
/*初始化链表*/  
envLinkHead = CreateEmptyLinklist();
```

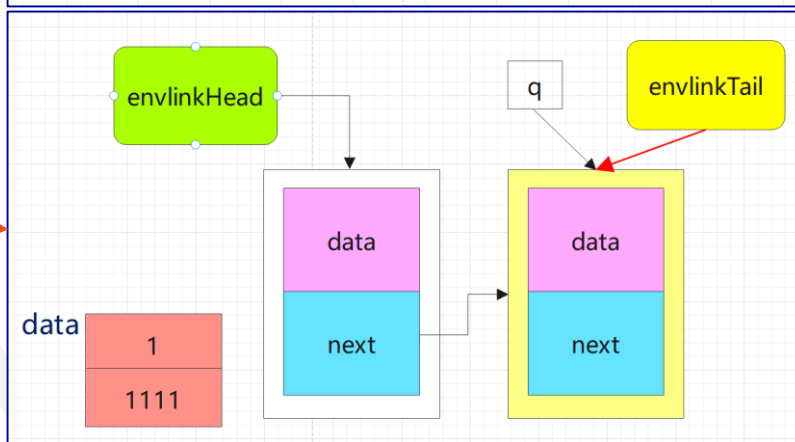
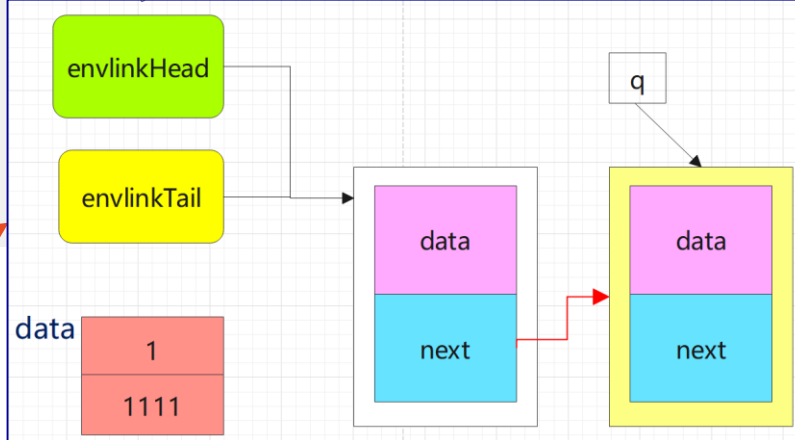




## 2.插入节点

```
int InsertLinknode(link_datatype x)
{
    linklist q = (linklist)malloc (sizeof (linknode));
    if (NULL == q)
    {
        printf ("InsertLinknode Error\n");
        return -1;
    }
    envlinkTail->next = q;
    envlinkTail = q;
    q->data = x;
    q->next = NULL;

    return 0;
}
```

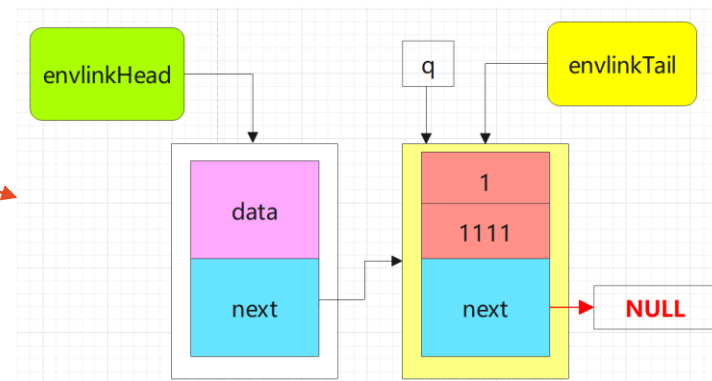
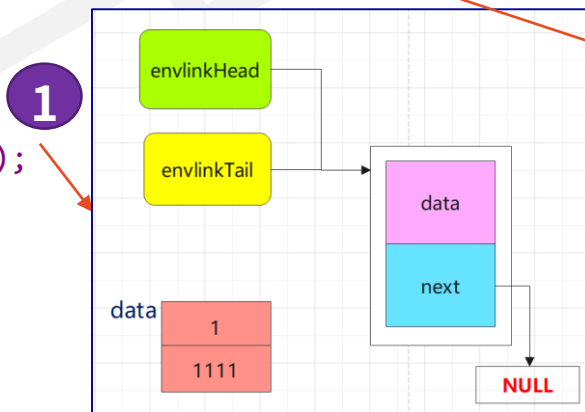


link\_datatype data;

data.msg\_type = 0x1;

memcpy(data.text, "1111", sizeof(data.text));

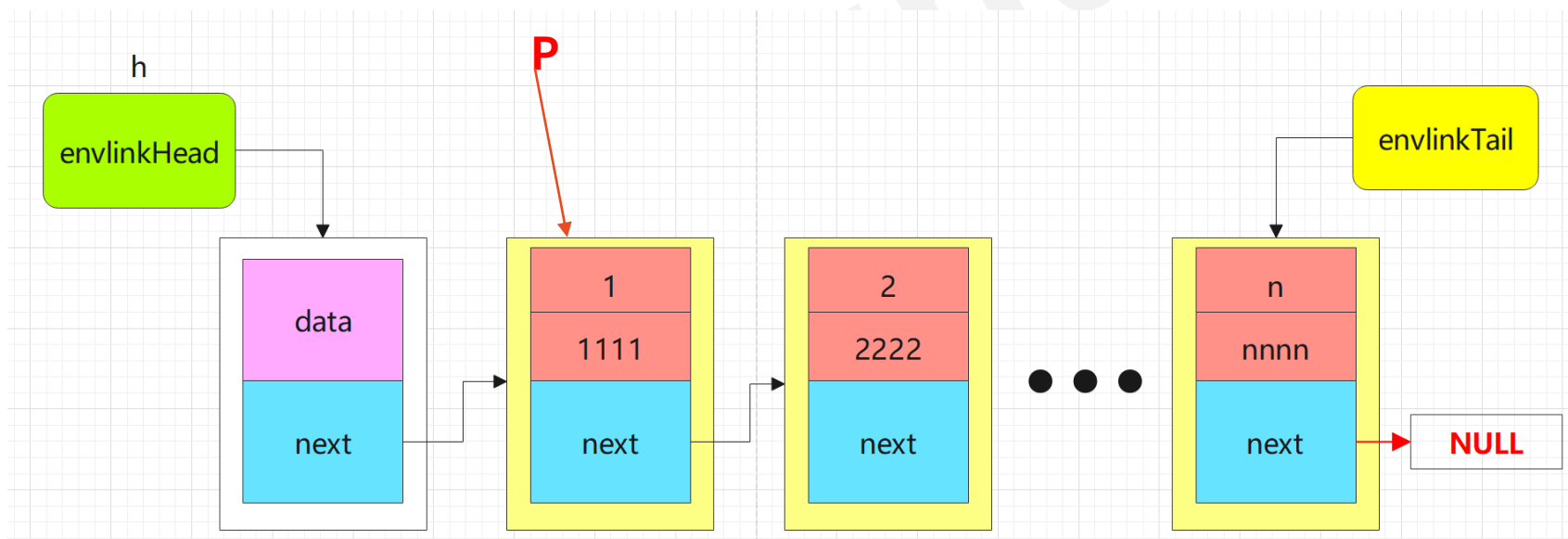
InsertLinknode (data)



# 3.遍历所有节点

```
void show_list(linklist h)
{
    int i = 0;
    linklist p;

    p = h->next;
    while(p != NULL)
    {
        printf("node[%d]:%s\n", i, p->data.text);
        i++;
        p = p->next;
    }
}
```

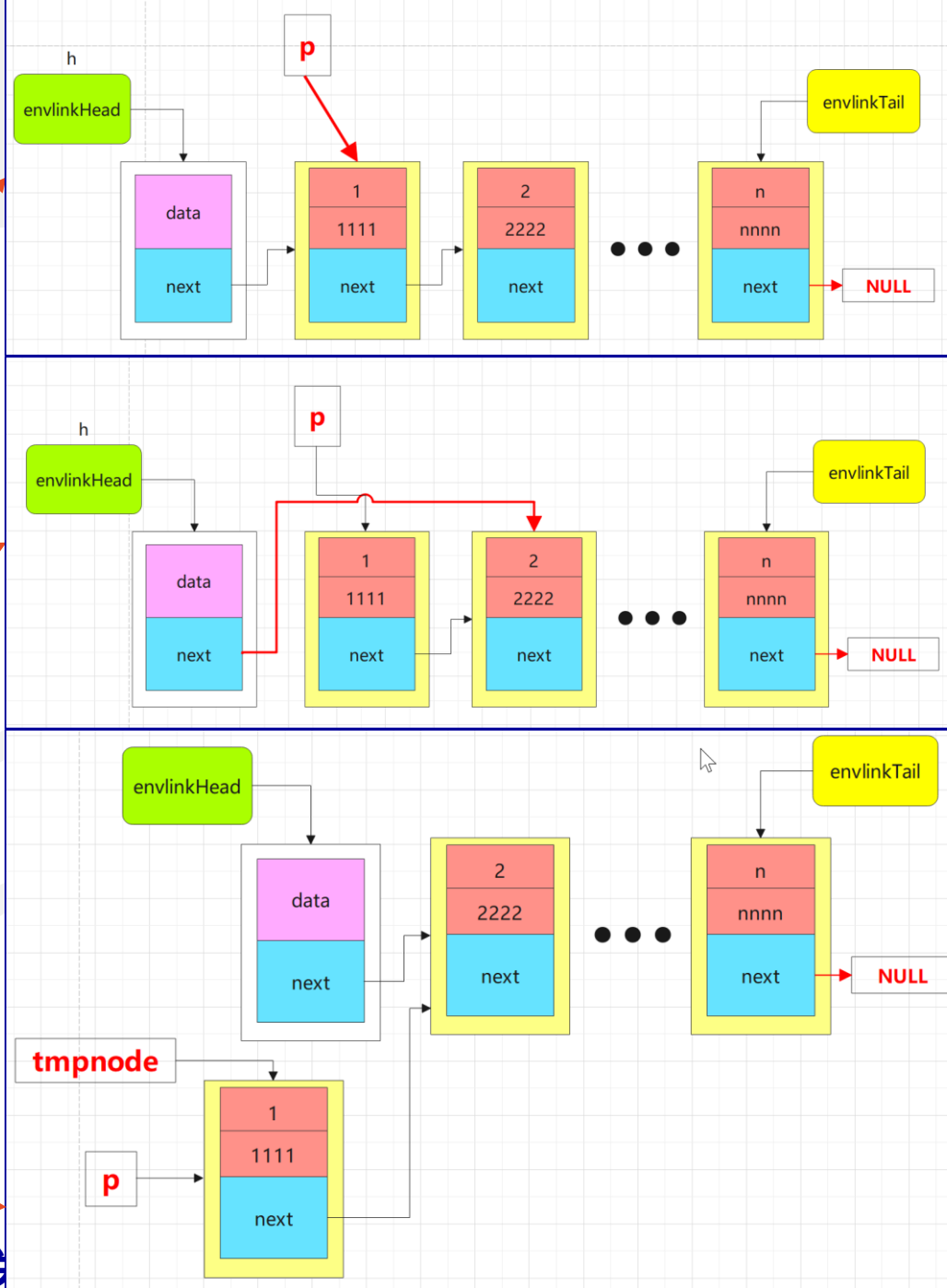


# 4.提取一个节点

```
int EmptyLinklist (linklist h)
{
    return NULL == h->next;
}

linklist GetLinknode(linklist h)
{
    if (1 == EmptyLinklist (h))
    {
        return NULL;
    }
    linklist p = h->next;
    h->next = p->next;
    if (p->next == NULL)
        envlinkTail = h;
    return p;
}
```

```
linklist tmpnode;
tmpnode = GetLinknode(envlinkHead);
free (tmpnode);
```



# 编译

- gcc main.c link\_list.c -o run

```
peng@ubuntu:~/work/test/list$ ls  
link_list.c link_list.h main.c  
peng@ubuntu:~/work/test/list$  
peng@ubuntu:~/work/test/list$  
peng@ubuntu:~/work/test/list$ gcc main.c link_list.c -o run
```