

OS lab9 Report

name: 刘乐奇

sid: 12011327

Ubuntu用户名: lynchrocket

1. 请将default_pmm.c中的85行le2page(le, page_link)宏展开，并简述le2page的工作原理（可以画图解释）

le2page() 宏定义如下

```
#define le2page(le, member) \
    to_struct((le), struct Page, member)
```

其中的 to_struct() 宏定义如下

```
#define to_struct(ptr, type, member) \
    ((type *)((char *) (ptr) - offsetof(type, member)))
```

其中的 offsetof() 宏定义如下

```
#define offsetof(type, member) \
    ((size_t) (&((type *)0)->member))
```

在 le2page(le, page_link) 中，将需要转换的 link entry le 和其对应的 Page 中的成员变量 page_link 传入（因为我们想要 le 所在的 Page 的首地址）。由于我们想将 link entry 转为 struct Page，所以传入 to_struct() 的 type 为 struct Page。在 to_struct() 中会将相应的 offset 减去。在 offsetof() 中，会通过 (type *)0 获得一个默认的该 type 的结构体指针，且该结构体的首地址为0。然后获得该结构体中相应 member 的偏移量并返回。

2. 请详细描述default_pmm.c中的default_alloc_pages和default_free_pages的功能与实现方式。

`default_alloc_pages()` 首先检测是否有足够的空闲空间来分配给要求的内存，如果没有就直接返回 NULL；否则从 `free_list` 开始逐个往后检查是否有 page 有足够的空间能分配给要求的内存。如果没找到，则返回 NULL；否则将该 page 的容量减去要求的内存，并设置相关的 flag bit。最后返回该 page。

`default_free_pages()` 首先从 `base` 开始往后找一共 `n` 大小的 page 内存（该内存不能是只供 kernel 的），然后将该 `n` 大小的 page 内存置为未分配。如果此时 `free_list` 之后没有 list entry，则直接指向该 `base`；若有，则找到地址大于 `base` 的 page，将 `base` 置于该 page 之前，或者若没有该 page，则置于 `free_list` 末尾。