CS302: Lab8 Report

Name：陆荻芸 SID：12011537

Answer 1

`le2page(le, page_link)`，宏定义了一个函数`to_struct((le),struct Page, member)`。其中 le 是`list_entry`类型的指针，`Page`是指针转换成的类型，`member`是 type 这个结构体类型的一个成员。`to_struct`调用了`offsetof()`，这个是计算`member`的偏移量的函数，他将 0 强转成 type 类型的指针，然后指向 member 成员，访问到的地址转换成`size_t`类型后就是这个成员相对于这个实例的偏移量大小。le 被转换成以字节为度量的地址，减去 offset 得到了想要转换为 page 的地址，然后再强转为 page 类型，就得到了一个 page。

Answer 2

`default_alloc_pages()`

这个函数实现了 firstfit 的内存分配算法。此时页面需求为 n，如果大于空闲的页数，就失败返回 null。如果空闲的够用的话，就遍历链表找到第一个空闲空间大于需求的空闲块。把该页从`page_link`里删除，更新新的头页的 property，减去需求，然后再加回`free_list`，同时更新空闲的页数，将旧页头清空。

`default_free_pages()`

这个函数实现了使用完毕的页面的释放。首先有一个 assert 判断这个 base 地址所在的页是否为预留，如果不是预留页，就无法再次 free。从 base 地址开始遍历插入的空闲页的大小，把标志位置 0 并把引用次数置 0、把空闲页数数量还回去。然后循环找到可合并的分区并一一还回去。如果欲释放内存在已有内存块之前，就从另一个方向还回去。