1. list_entry - get the struct for this entry.

ptr:	the *&struct list_head* pointer.
type:	the type of the struct this is embedded in.
member:	the name of the list_head within the struct.

#define *list_entry*(*ptr*, *type*, *member*)
    container_of(*ptr*, *type*, *member*)

2. list_prev_entry - get the prev element in list.

pos:	the type * to cursor.
member:	the name of the list_head within the struct.

#define *list_prev_entry*(*pos*, *member*)
    list_entry((*pos*)→*member*.*prev*, *typeof*(∗(*pos*)), *member*)

3. list_next_entry - get the next element in list.

pos:	the type * to cursor.
member:	the name of the list_head within the struct.

#define *list_next_entry*(*pos*, *member*)
    list_entry((*pos*)→*member*.*next*, *typeof*(∗(*pos*)), *member*)

4. list_last_entry - get the last element from a list.

ptr:	the list head to take the element from.
type:	the type of the struct this is embedded in.
member:	the name of the list_head within the struct.

#define *list_last_entry*(*ptr*, *type*, *member*)
    list_entry((*ptr*)→*prev*, *type*, *member*)

5. list_first_entry - get the first element from a list.

ptr:	the list head to take the element from.
type:	the type of the struct this is embedded in.
member:	the name of the list_head within the struct.

#define *list_first_entry*(*ptr*, *type*, *member*)
    list_entry((*ptr*)→*next*, *type*, *member*)

6. list_entry_is _head - test if the entry points to the head of the list.

pos:	the type * to cursor.
head:	the head for your list.
member:	the name of the list_head within the struct.

#define *list_entry_is_head*(*pos*, *head*, *member*)
    (&*pos*→*member* ≡ (*head*))

7. list_for_each_entry - iterate over list of given type.

pos:	the type * to cursor.
head:	the head for your list.
member:	the name of the list_head within the struct.

#define *list_for_each_entry*(*pos*, *head*, *member*)
    for (*pos* = *list_first_entry*(*head*, *typeof*(∗*pos*), *member*);
    ¬*list_entry_is_head*(*pos*, *head*, *member*);
    pos = *list_next_entry*(*pos*, *member*))

8. list_for_each_entry_reverse - iterate backwards over list of given type.

pos:	the type * to cursor.
head:	the head for your list.
member:	the name of the list_head within the struct.

```
#define list_for_each_entry_reverse(pos, head, member)
        for (pos = list_last_entry(head, typeof(*pos), member);
        ¬list_entry_is_head(pos, head, member);
        pos = list_prev_entry(pos, member))
```

9. list_for_each_entry_safe - iterate over list of given type safe against removal of list entry.

pos:	the type * to use as a loop cursor.
n:	another type * to use as temporary storage.
head:	the head for your list.
member:	the name of the list_head within the struct.

```
#define list_for_each_entry_safe(pos, n, head, member)
        for (pos = list_first_entry(head, typeof(*pos), member),
        n = list_next_entry(pos, member);
        ¬list_entry_is_head(pos, head, member);
        pos = n, n = list_next_entry(n, member))
```

10. Index.

C