glibc 中 __libc_free 函数的源码

```
void
__libc_free (void *mem)
{
  mstate ar_ptr;
  mchunkptr p;                          /* chunk corresponding to mem */

  void (*hook) (void *, const void *)
    = atomic_forced_read (__free_hook); /* ? */
  if (__builtin_expect (hook != NULL, 0))
    {
      (*hook)(mem, RETURN_ADDRESS (0)); /* ? */
      return;
    }

  if (mem == 0)                              /* free(0) has no effect */
    return;

  p = mem2chunk (mem);

  if (chunk_is_mmapped (p))                       /* release mmapped memory.
*/
    {
      /* see if the dynamic brk/mmap threshold needs adjusting */
      if (!mp_.no_dyn_threshold
          && p->size > mp_.mmap_threshold
          && p->size <= DEFAULT_MMAP_THRESHOLD_MAX)
        {
          mp_.mmap_threshold = chunksize (p);
          mp_.trim_threshold = 2 * mp_.mmap_threshold;
          LIBC_PROBE (memory_mallopt_free_dyn_thresholds, 2,
                      mp_.mmap_threshold, mp_.trim_threshold);
        }
      munmap_chunk (p);
      return;
    }

  ar_ptr = arena_for_chunk (p);
  _int_free (ar_ptr, p, 0);
}
libc_hidden_def (__libc_free)
```