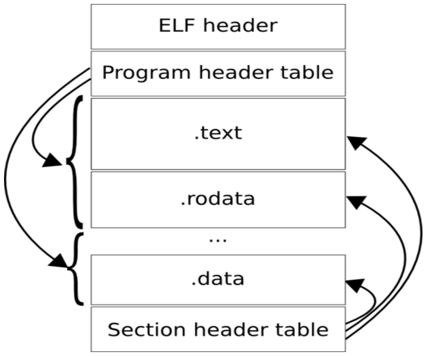


e_entry This member gives the virtual address to which the system first transfers control, thus starting the process. If the file has no associated entry point, this member holds zero.

e_phoff This member holds the program header table's file offset in bytes. If the file has no program header table, this member holds zero.

e_shoff This member holds the section header table's file offset in bytes. If the file has no section header table, this member holds zero.

e_phnum This member holds the number of entries in the program header table.

P_type

PT_LOAD The array element specifies a loadable segment, described by *p_filesz* and *p_memsz*. The bytes from the file are mapped to the beginning of the memory segment. If the segment's memory size *p_memsz* is larger than the file size *p_filesz*, the "extra" bytes are defined to hold the value 0 and to follow the segment's initialized area. The file size may not be larger than the memory size. Loadable segment entries in the program header table appear in ascending order, sorted on the *p_vaddr* member.

p_offset This member holds the offset from the beginning of the file at which the first byte of the segment resides.

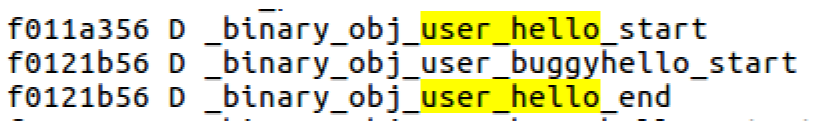
p_vaddr This member holds the virtual address at which the first byte of the segment resides in memory.

p_filesz This member holds the number of bytes in the file image of the segment. It may be zero.

p_memsz This member holds the number of bytes in the memory image of the segment. It may be zero.

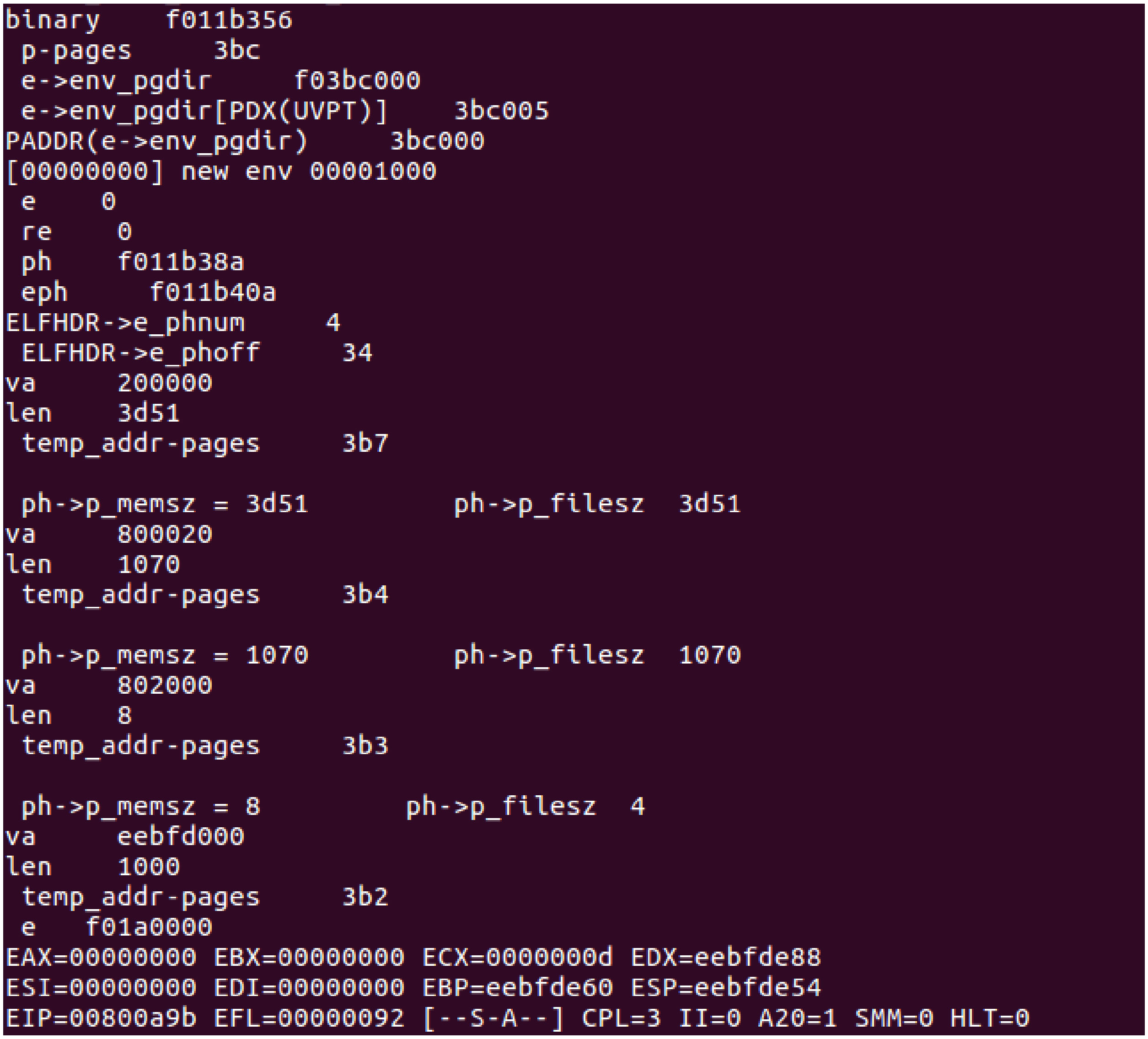
The reason why `p_memsz` is greater than (or equal to) `p_filesz` is that a loadable segment may contain a `.bss` section, which contains uninitialized data.

Boot_alloc uses:
We use this function whenever we want to use memory, its uses:
Allocating one page to the kern_dirc page, and we filled it with data later
Allocating one page to the page metadata array, and fill it with info
Allocating one page to the env metadata array, and fill it with info
And after that we should use only page_alloc



```
f011a356 D _binary_obj_user_hello_start
f0121b56 D _binary_obj_user_buggyhello_start
f0121b56 D _binary_obj_user_hello_end
```

0x00800e25



```
binary    f011b356
 p-pages     3bc
 e->env_pgdir     f03bc000
 e->env_pgdir[PDX(UVPT)]    3bc005
PADDR(e->env_pgdir)     3bc000
[00000000] new env 00001000
 e    0
 re    0
 ph    f011b38a
 eph     f011b40a
ELFHDR->e_phnum     4
 ELFHDR->e_phoff     34
va      200000
len     3d51
 temp_addr-pages     3b7

 ph->p_memsz = 3d51         ph->p_filesz  3d51
va      800020
len     1070
 temp_addr-pages     3b4

 ph->p_memsz = 1070         ph->p_filesz  1070
va      802000
len     8
 temp_addr-pages     3b3

 ph->p_memsz = 8         ph->p_filesz  4
va      eebfd000
len     1000
 temp_addr-pages     3b2
 e   f01a0000
EAX=00000000 EBX=00000000 ECX=0000000d EDX=eebfde88
ESI=00000000 EDI=00000000 EBP=eebfde60 ESP=eebfde54
EIP=00800a9b EFL=00000092 [--S-A--] CPL=3 II=0 A20=1 SMM=0 HLT=0
```