1. 1个, 管理该进程的的所有页表
2. 管理一块连续的 Virtual Memory
3. 访问不在页表中的地址, 或者在usermode下访问非usermode可操作的页表
4. 访问的虚拟地址内容不在内存中，需要swap in

```
45          if (swap_init_ok) {
46              struct Page *page = NULL;
47              swap_in(mm, addr, &page);  //According to the mm AND addr, try
48                                         //to load the content of right disk pag
49                                         //    into the memory which page manage
50              page_insert(mm->pgdir, page, addr, perm);  //According to the mm,
51                                                         //addr AND page, setup the
52                                                         //map of phy addr <--->
53                                                         //logical addr
54              swap_map_swappable(mm, addr, page,
55                                 1);  //make the page swappable.
56              page->pra_vaddr = addr;
```

5. swapin: 访问的虚拟地址内容不在内存中 swapout: page塞满了需要腾出空间来