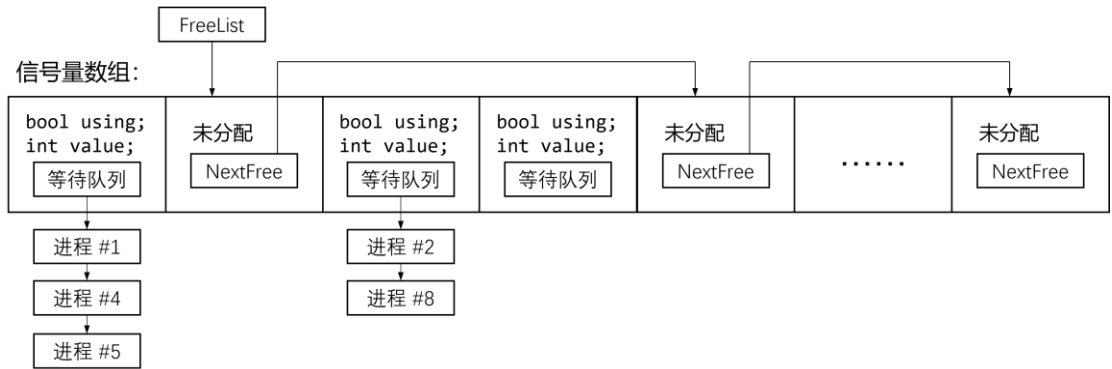


第 3 次操作系统实验报告

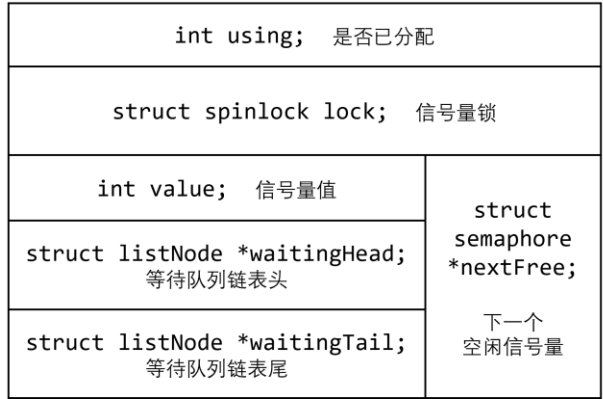
161810129 董世晨

一、 信号量

1. 思路介绍



Semaphore结构体:



```
struct semaphore {
    int using;
    struct spinlock lock;
    union {
        struct {
            int value;
            struct listNode *waitingHead;
            struct listNode *waitingTail;
        };
        struct semaphore *nextFree;
    };
};
sems[NSEM];
```

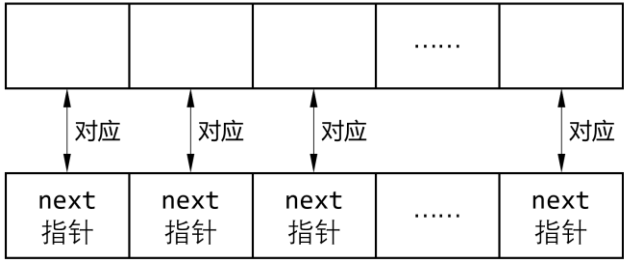
图注：纵向堆叠表示struct，横向堆叠表示union。

上图包含了我认为设计中比较关键的地方，相信别人看后应该比较容易复现我的方案。接下来我介绍一下设计中需要注意的细节。

2. 如何实现等待队列

首先，我们不希望出现饥饿的情况，队列结构可以符合这个要求。另外观察到：每个进程或者不等待，或者正在等待一个信号量，不可能出现同时等待多个信号量的情况。最简单的方式是在PCB中新增一个字段，记录等待队列列表的下一个。

ptable.proc数组:



proclist数组:

```
struct listNode {
    struct listNode *next;
} proclist[NPROC];
```

但是由于存在奇怪的要求，PCB中不能新增字段，因此我开辟了一个和`ptable.proc`大小相同的数组`proclist`，其中元素和`ptable.proc`中的元素按下标一一对应，如上图所示。例如，`proclist[5]`对应了`ptable.proc`中的下标为5的PCB，注意不是和PID为5的进程对应。这样子，`listNode`作为链表的结点，作为进程的替身，被加入到了等待队列中。

3. 锁

除了每一个信号量有一个细粒度锁来保证wait和signal时不发生数据竞争，还需要一个大粒度锁来保护整个信号量数组在索引、分配和释放信号量时不发生数据竞争。另外，在修改PCB内容和进入调度器时，需要锁定ptable，并释放已获取的锁。

比起为什么需要加锁，我觉得更值得思考的是**为什么不需要加锁**。在我的设计中，不需要对proclist加锁，也就是说，**如果多个进程同时执行wait_sem，但wait的具体信号量不同，它们能并发执行**。只有wait_sem和wait_signal这两个函数读取和修改对proclist中的内容，下面讨论它们并发执行的三种情况：

- 1) 两个进程同时执行wait_sem: 由于ptable.proc和prolist中的元素一一对应, 这两个进程对应的proclist元素必然不同, 因此不会发生冲突。
- 2) 两个进程同时执行signal_sem: 如果signal的信号量相同, 信号量的小粒度锁保证了它们不会同时执行; 如果signal的信号量不同, 由于一个进程不可能同时wait两个信号量, 因此这两个进程唤醒的进程必然不同, 它们对应的proclist中的元素必然不同, 不会发生冲突。
- 3) 两个进程分别执行wait_sem和signal_sem: 如果操作的信号量相同, 信号量的小粒度锁会保证它们不会同时执行; 如果操作的信号量不同, 由于正在wait的进程不可能正在执行别的代码, 因此signal_sem中唤醒的进程必然不是执行wait_sem的进程, 它们对应的proclist中的元素必然不同, 不会发生冲突。

至此，在任何情况下都不会有进程同时读取和修改proclist中的同一个元素，不需要给proclist加锁证毕。

4. 运行结果

[illegible]

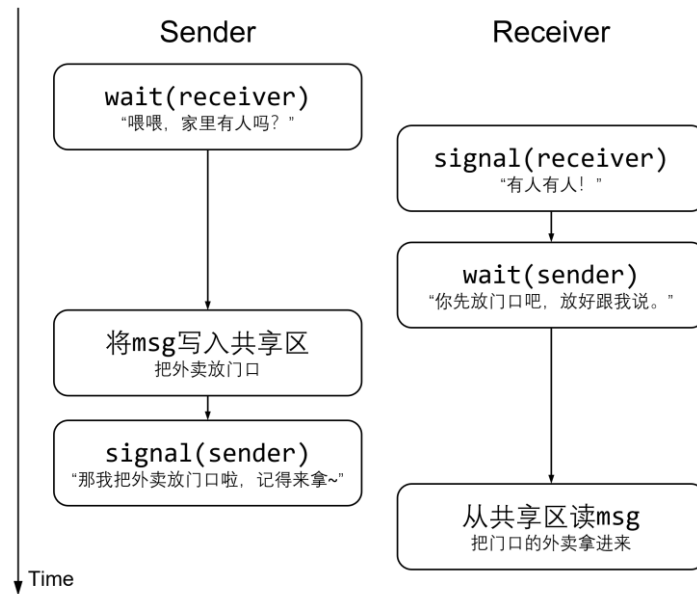
二、 消息传递

1. 思路介绍

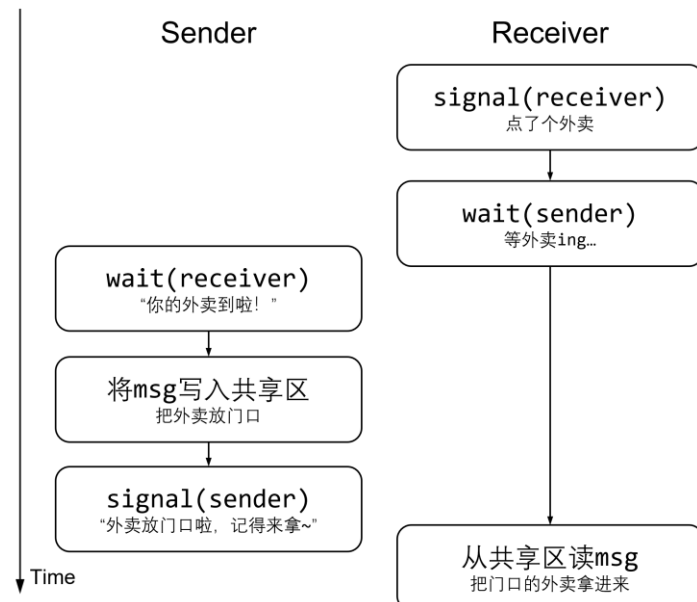
本质上，对于每个进程来说，这是多个生产者（消息发送者）和单个消费者（消息接收者），且没有缓冲区（发送者必须等到消息被接受后才能返回）的生产者-消费者问题。因此我参考了多生产者-单消费者的解决方案：使用两个信号量，一个表示receiver是否等待，一个表示正在等待的sender的数量。

上面这段话可能比较抽象，下面让我用一个“点外卖”的例子来解释：

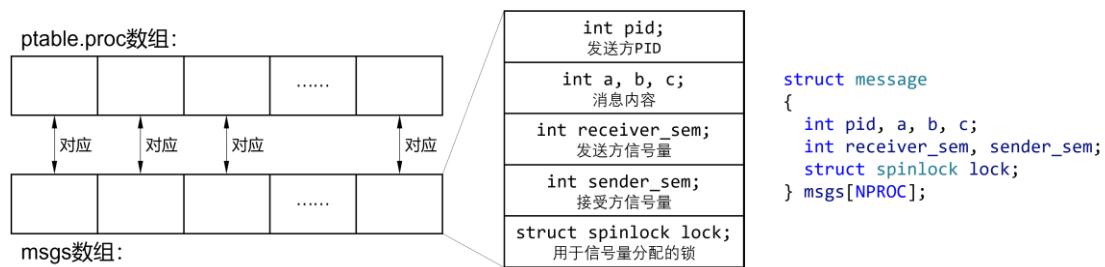
当Sender先执行时：



当Receiver先执行时：



显然，每个进程都可以作为接收者，因此每个进程都需要两个信号量来控制、和一块共享内存区用于存放消息内容。和信号量一样，我开辟了一个大小和`ptable.proc`相同的数组`msgs`，其中元素和`ptable.proc`中的元素按下标一一对应，如下图所示：



事实上，前文所述的send和receive的流程已经可以保证收发双方不会在共享内存区中发生数据竞争，那为什么在message结构体中还需要一个锁呢？这要从信号量的分配说起。

题目要求信号量必须在使用时分配，不能提前分配，因此，我用-1表示信号量没有被分配，在每次wait或signal信号量前检查信号量有没有被分配，如果没有，则立即分配一个。令人遗憾的是，分配信号量的过程可能会发生冲突，即两个sender可能同时判断信号量没有被分配，然后同时分配信号量。所以必须要使用一个锁来保护分配过程。又由于分配信号量的过程很快，使用自旋锁完全可以满足要求。

2. 实现细节

由于很多地方需要检查并分配信号量，我对alloc_sem做了简单的封装，写了一个异常安全的信号量分配函数：要么分配成功，要么分配失败且没有任何副作用。

另外在进程退出时，即wait函数内，需要释放可能已经分配的信号量。由于dealloc_sem内部需要获取ptable锁，不能直接在wait函数已经持有ptable锁时调用dealloc_sem来释放信号量。我的解决方案是：当在wait函数内调用dealloc_sem时不让他获取ptable锁，可以通过一个函数参数来控制，具体见代码中的dealloc_sem_lk(v, 0)辅助函数。

3. 运行结果

```
$ msgtest
child(sender,child=3): 24(23,27) 25(23,28) 26(23,29) 27(23,328(23,31) 29(23,32) 30(23,33) 31(23,34) 32(23,35) 33(23,36) 34(23,37) 35(23,38) 36(23,39) 38(23,41) 39(23,42) 40(23,43) 41(23,44) 42(23,45) 43(23,46) 44(23,47) 45(23,48) 46(23,49) 47(23,50) 48(23,51) 49(23,52) 50(23,53) 51(23,54) 52(23,55) 53(23,56) 54(23,57) 55(23,58) 56(23,59) 57(23,60) 58(23,61) 59(23,62) 60(23,63) 61(23,64) 62(23,65) 63(23,66) 64(23,67) 65(23,68) 66(23,69) 67(23,70) 68(23,71) 69(23,72) 70(23,73) 71(23,74) 72(23,75) 73(23,76) 74(23,77) 75(23,78) 76(23,79) 77(23,80) 78(23,81) 79(23,82) 80(23,83) 81(23,84) 82(23,85) 83(23,86) 84(23,87) 85(23,88) 86(23,89) 87(23,90) 88(23,91) 89(23,92) 90(23,93) 91(23,94) 92(23,95) 93(23,96) 94(23,97) 95(23,98) 96(23,99) 97(23,100) 98(23,101) 99(23,102) 100(23,103) 101(23,104) 102(23,105) 103(23,106) 104(23,107) 105(23,108) 106(23,109) 107(23,110) 108(23,111) 109(23,112) 110(23,113) 111(23,114) 112(23,115) 113(23,116) 114(23,117) 115(23,118) 116(23,119) 117(23,120) 118(23,121) 119(23,122) 120(23,123) 121(23,124) 122(23,125) 123(23,126) 124(23,127) 125(23,128) 126(23,129) 127(23,130) 128(23,131) 129(23,132) 130(23,133) 131(23,134) 132(23,135) 133(23,136) 134(23,137) 135(23,138) 136(23,139) 137(23,140) 138(23,141) 139(23,142) 140(23,143) 141(23,144) 142(23,145) 143(23,146) 144(23,147) 145(23,148) 146(23,149) 147(23,150) 148(23,151) 149(23,152) 150(23,153) 151(23,154) 152(23,155) 153(23,156) 154(23,157) 155(23,158) 156(23,159) 157(23,160) 158(23,161) 159(23,162) 160(23,163) 161(23,164) 162(23,165) 163(23,166) 164(23,167) 165(23,168) 166(23,169) 167(23,170) 168(23,171) 169(23,172) 170(23,173) 171(23,174) 172(23,175) 173(23,176) 174(23,177) 175(23,178) 176(23,179) 177(23,180) 178(23,181) 179(23,182) 180(23,183) 181(23,184) 182(23,185) 183(23,186) 184(23,187) 185(23,188) 186(23,189) 187(23,190) 188(23,191) 189(23,192) 190(23,193) 191(23,194) 192(23,195) 193(23,196) 194(23,197) 195(23,198) 196(23,199) 197(23,200) 198(23,201) 199(23,202) 200(23,203) 201(23,204) 202(23,205) 203(23,206) 204(23,207) 205(23,208) 206(23,209) 207(23,210) 208(23,211) 209(23,212) 210(23,213) 211(23,214) 212(23,215) 213(23,216) 214(23,217) 215(23,218) 216(23,219) 217(23,220) 218(23,221) 219(23,222) 220(23,223) 221(23,224) 222(23,225) 223(23,226) 224(23,227) 225(23,228) 226(23,229) 227(23,230) 228(23,231) 229(23,232) 230(23,233) 231(23,234) 232(23,235) 233(23,236) 234(23,237) 235(23,238) 236(23,239) 237(23,240) 238(23,241) 239(23,242) 240(23,243) 241(23,244) 242(23,245) 243(23,246) 244(23,247) 245(23,248) 246(23,249) 247(23,250) 248(23,251) 249(23,252) 250(23,253) 251(23,254) 252(23,255) 253(23,256) 254(23,257) 255(23,258) 256(23,259) 257(23,260) 258(23,261) 259(23,262) 260(23,263) 261(23,264) 262(23,265) 263(23,266) 264(23,267) 265(23,268) 266(23,269) 267(23,270) 268(23,271) 269(23,272) 270(23,273) 271(23,274) 272(23,275) 273(23,276) 274(23,277) 275(23,278) 276(23,279) 277(23,280) 278(23,281) 279(23,282) 280(23,283) 281(23,284) 282(23,285) 283(23,286) 284(23,287) 285(23,288) 286(23,289) 287(23,290) 288(23,291) 289(23,292) 290(23,293) 291(23,294) 292(23,295) 293(23,296) 294(23,297) 295(23,298) 296(23,299) 297(23,300) 298(23,301) 299(23,302) 300(23,303) 301(23,304) 302(23,305) 303(23,306) 304(23,307) 305(23,308) 306(23,309) 307(23,310) 308(23,311) 309(23,312) 310(23,313) 311(23,314) 312(23,315) 313(23,316) 314(23,317) 315(23,318) 316(23,319) 317(23,320) 318(23,321) 319(23,322) 320(23,323) 321(23,324) 322(23,325) 323(23,326) 324(23,327) 325(23,328) 326(23,329) 327(23,330) 328(23,331) 329(23,332) 330(23,333) 331(23,334) 332(23,335) 333(23,336) 334(23,337) 335(23,338) 336(23,339) 337(23,340) 338(23,341) 339(23,342) 340(23,343) 341(23,344) 342(23,345) 343(23,346) 344(23,347) 345(23,348) 346(23,349) 347(23,350) 348(23,351) 349(23,352) 350(23,353) 351(23,354) 352(23,355) 353(23,356) 354(23,357) 355(23,358) 356(23,359) 357(23,360) 358(23,361) 359(23,362) 360(23,363) 361(23,364) 362(23,365) 363(23,366) 364(23,367) 365(23,368) 366(23,369) 367(23,370) 368(23,371) 369(23,372) 370(23,373) 371(23,374) 372(23,375) 373(23,376) 374(23,377) 375(23,378) 376(23,379) 377(23,380) 378(23,381) 379(23,382) 380(23,383) 381(23,384) 382(23,385) 383(23,386) 384(23,387) 385(23,388) 386(23,389) 387(23,390) 388(23,391) 389(23,392) 390(23,393) 391(23,394) 392(23,395) 393(23,396) 394(23,397) 395(23,398) 396(23,399) 397(23,400) 398(23,401) 399(23,402) 400(23,403) 401(23,404) 402(23,405) 403(23,406) 404(23,407) 405(23,408) 406(23,409) 407(23,410) 408(23,411) 409(23,412) 410(23,413) 411(23,414) 412(23,415) 413(23,416) 414(23,417) 415(23,418) 416(23,419) 417(23,420) 418(23,421) 419(23,422) 420(23,423) 421(23,424) 422(23,425) 423(23,426) 424(23,427) 425(23,428) 426(23,429) 427(23,430) 428(23,431) 429(23,432) 430(23,433) 431(23,434) 432(23,435) 433(23,436) 434(23,437) 435(23,438) 436(23,439) 437(23,440) 438(23,441) 439(23,442) 440(23,443) 441(23,444) 442(23,445) 443(23,446) 444(23,447) 445(23,448) 446(23,449) 447(23,450) 448(23,451) 449(23,452) 450(23,453) 451(23,454) 452(23,455) 453(23,456) 454(23,457) 455(23,458) 456(23,459) 457(23,460) 458(23,461) 459(23,462) 460(23,463) 461(23,464) 462(23,465) 463(23,466) 464(23,467) 465(23,468) 466(23,469) 467(23,470) 468(23,471) 469(23,472) 470(23,473) 471(23,474) 472(23,475) 473(23,476) 474(23,477) 475(23,478) 476(23,479) 477(23,480) 478(23,481) 479(23,482) 480(23,483) 481(23,484) 482(23,485) 483(23,486) 484(23,487) 485(23,488) 486(23,489) 487(23,490) 488(23,491) 489(23,492) 490(23,493) 491(23,494) 492(23,495) 493(23,496) 494(23,497) 495(23,498) 496(23,499) 497(23,500) 498(23,501) 499(23,502) 500(23,503) 501(23,504) 502(23,505) 503(23,506) 504(23,507) 505(23,508) 506(23,509) 507(23,510) 508(23,511) 509(23,512) 510(23,513) 511(23,514) 512(23,515) 513(23,516) 514(23,517) 515(23,518) 516(23,519) 517(23,520) 518(23,521) 519(23,522) 520(23,523) 521(23,524) 522(23,525) 523(23,526) 524(23,527) 525(23,528) 526(23,529) 527(23,530) 528(23,531) 529(23,532) 530(23,533) 531(23,534) 532(23,535) 533(23,536) 534(23,537) 535(23,538) 536(23,539) 537(23,540) 538(23,541) 539(23,542) 540(23,543) 541(23,544) 542(23,545) 543(23,546) 544(23,547) 545(23,548) 546(23,549) 547(23,550) 548(23,551) 549(23,552) 550(23,553) 551(23,554) 552(23,555) 553(23,556) 554(23,557) 555(23,558) 556(23,559) 557(23,560) 558(23,561) 559(23,562) 560(23,563) 561(23,564) 562(23,565) 563(23,566) 564(23,567) 565(23,568) 566(23,569) 567(23,570) 568(23,571) 569(23,572) 570(23,573) 571(23,574) 572(23,575) 573(23,576) 574(23,577) 575(23,578) 576(23,579) 577(23,580) 578(23,581) 579(23,582) 580(23,583) 581(23,584) 582(23,585) 583(23,586) 584(23,587) 585(23,588) 586(23,589) 587(23,590) 588(23,591) 589(23,592) 590(23,593) 591(23,594) 592(23,595) 593(23,596) 594(23,597) 595(23,598) 596(23,599) 597(23,600) 598(23,601) 599(23,602) 600(23,603) 601(23,604) 602(23,605) 603(23,606) 604(23,607) 605(23,608) 606(23,609) 607(23,610) 608(23,611) 609(23,612) 610(23,613) 611(23,614) 612(23,615) 613(23,616) 614(23,617) 615(23,618) 616(23,619) 617(23,620) 618(23,621) 619(23,622) 620(23,623) 621(23,624) 622(23,625) 623(23,626) 624(23,627) 625(23,628) 626(23,629) 627(23,630) 628(23,631) 629(23,632) 630(23,633) 631(23,634) 632(23,635) 633(23,636) 634(23,637) 635(23,638) 636(23,639) 637(23,640) 638(23,641) 639(23,642) 640(23,643) 641(23,644) 642(23,645) 643(23,646) 644(23,647) 645(23,648) 646(23,649) 647(23,650) 648(23,651) 649(23,652) 650(23,653) 651(23,654) 652(23,655) 653(23,656) 654(23,657) 655(23,658) 656(23,659) 657(23,660) 658(23,661) 659(23,662) 660(23,663) 661(23,664) 662(23,665) 663(23,666) 664(23,667) 665(23,668) 666(23,669) 667(23,670) 668(23,671) 669(23,672) 670(23,673) 671(23,674) 672(23,675) 673(23,676) 674(23,677) 675(23,678) 676(23,679) 677(23,680) 678(23,681) 679(23,682) 680(23,683) 681(23,684) 682(23,685) 683(23,686) 684(23,687) 685(23,688) 686(23,689) 687(23,690) 688(23,691) 689(23,692) 690(23,693) 691(23,694) 692(23,695) 693(23,696) 694(23,697) 695(23,698) 696(23,699) 697(23,700) 698(23,701) 699(23,702) 700(23,703) 701(23,704) 702(23,705) 703(23,706) 704(23,707) 705(23,708) 706(23,709) 707(23,710) 708(23,711) 709(23,712) 710(23,713) 711(23,714) 712(23,715) 713(23,716) 714(23,717) 715(23,718) 716(23,719) 717(23,720) 718(23,721) 719(23,722) 720(23,723) 721(23,724) 722(23,725) 723(23,726) 724(23,727) 725(23,728) 726(23,729) 727(23,730) 728(23,731) 729(23,732) 730(23,733) 731(23,734) 732(23,735) 733(23,736) 734(23,737) 735(23,738) 736(23,739) 737(23,740) 738(23,741) 739(23,742) 740(23,743) 741(23,744) 742(23,745) 743(23,746) 744(23,747) 745(23,748) 746(23,749) 747(23,750) 748(23,751) 749(23,752) 750(23,753) 751(23,754) 752(23,755) 753(23,756) 754(23,757) 755(23,758) 756(23,759) 757(23,760) 758(23,761) 759(23,762) 760(23,763) 761(23,764) 762(23,765) 763(23,766) 764(23,767) 765(23,768) 766(23,769) 767(23,770) 768(23,771) 769(23,772) 770(23,773) 771(23,774) 772(23,775) 773(23,776) 774(23,777) 775(23,778) 776(23,779) 777(23,780) 778(23,781) 779(23,782) 780(23,783) 781(23,784) 782(23,785) 783(23,786) 784(23,787) 785(23,788) 786(23,789) 787(23,790) 788(23,791) 789(23,792) 790(23,793) 791(23,794) 792(23,795) 793(23,796) 794(23,797) 795(23,798) 796(23,799) 797(23,800) 798(23,801) 799(23,802) 800(23,803) 801(23,804) 802(23,805) 803(23,806) 804(23,807) 805(23,808) 806(23,809) 807(23,810) 808(23,811) 809(23,812) 810(23,813) 811(23,814) 812(23,815) 813(23,816) 814(23,817) 815(23,818) 816(23,819) 817(23,820) 818(23,821) 819(23,822) 820(23,823) 821(23,824) 822(23,825) 823(23,826) 824(23,827) 825(23,828) 826(23,829) 827(23,830) 828(23,831) 829(23,832) 830(23,833) 831(23,834) 832(23,835) 833(23,836) 834(23,837) 835(23,838) 836(23,839) 837(23,840) 838(23,841) 839(23,842) 840(23,843) 841(23,844) 842(23,845) 843(23,846) 844(23,847) 845(23,848) 846(23,849) 847(23,850) 848(23,851) 849(23,852) 850(23,853) 851(23,854) 852(23,855) 853(23,856) 854(23,857) 855(23,858) 856(23,859) 857(23,860) 858(23,861) 859(23,862) 860(23,863) 861(23,864) 862(23,865) 863(23,866) 864(23,867) 865(23,868) 866(23,869) 867(23,870) 868(23,871) 869(23,872) 870(23,873) 871(23,874) 872(23,875) 873(23,876) 874(23,877) 875(23,878) 876(23,879) 877(23,880) 878(23,881) 879(23,882) 880(23,883) 881(23,884) 882(23,885) 883(23,886) 884(23,887) 885(23,888) 886(23,889) 887(23,890) 888(23,891) 889(23,892) 890(23,893) 891(23,894) 892(23,895) 893(23,896) 894(23,897) 895(23,898) 896(23,899) 897(23,900) 898(23,901) 899(23,902) 900(23,903) 901(23,904) 902(23,905) 903(23,906) 904(23,907) 905(23,908) 906(23,909) 907(23,910) 908(23,911) 909(23,912) 910(23,913) 911(23,914) 912(23,915) 913(23,916) 914(23,917) 915(23,918) 916(23,919) 917(23,920) 918(23,921) 919(23,922) 920(23,923) 921(23,924) 922(23,925) 923(23,926) 924(23,927) 925(23,928) 926(23,929) 927(23,930) 928(23,931) 929(23,932) 930(23,933) 931(23,934) 932(23,935) 933(23,936) 934(23,937) 935(23,938) 936(23,939) 937(23,940) 938(23,941) 939(23,942) 940(23,943) 941(23,944) 942(23,945) 943(23,946) 944(23,947) 945(23,948) 946(23,949) 947(23,950) 948(23,951) 949(23,952) 950(23,953) 951(23,954) 952(23,955) 953(23,956) 954(23,957) 955(23,958) 956(23,959) 957(23,960) 958(23,961) 959(23,962) 960(23,963) 961(23,964) 962(23,965) 963(23,966) 964(23,967) 965(23,968) 966(23,969) 967(23,970) 968(23,971) 969(23,972) 970(23,973) 971(23,974) 972(23,975) 973(23,976) 974(23,977) 975(23,978) 976(23,979) 977(23,980) 978(23,981) 979(23,982) 980(23,983) 981(23,984) 982(23,985) 983(23,986) 984(23,987) 985(23,988) 986(23,989) 987(23,990) 988(23,991) 989(23,992) 990(23,993) 991(23,994) 992(23,995) 993(23,996) 994(23,997) 995(23,998) 996(23,999) 997(24,0) 998(24
```