



Pontifícia Universidade Católica de Campinas
Faculdade de Engenharia de Computação -
FECOMP

Sistemas Operacionais A – Relatório Experimento 5

Beatriz Morelatto Lorente RA: 18071597

Cesar Marrote Manzano RA: 18051755

Fabricio Silva Cardoso RA: 18023481

Pedro Ignácio Trevisan RA: 18016568

Sumário

1.Introdução.....	3
2.Respostas das perguntas.....	4
3.Resultados da execução do primeiro programa.....	5
4.Resultados da execução do segundo programa.....	9
5.Análise dos Resultados.....	14
6.Conclusão.....	24

Introdução

O experimento realizado permitiu o melhor entendimento de todos os conceitos adquiridos ao longo do semestre, como: criação de processos filhos, fila de mensagens, memória compartilhada, semáforos e threads. Foram feitos dois programas diferentes, visando a resolução do problema clássico do barbeiro dorminhoco.

No primeiro programa, era necessário o uso de processos filhos, fila de mensagens e memória compartilhada. O número de cadeiras disponíveis era igual a sete, e era necessário inicializar dois processos barbeiros e vinte processos clientes. Um processo cliente, envia uma mensagem ao barbeiro, com uma string de tamanho e números variáveis (entre 2 e 1023) e o barbeiro devolvia a mensagem com o resultado do serviço prestado. Caso o cliente não fosse atendido, era necessário mostrar o seu número. Caso contrário, era necessário imprimir o resultado da string que seria, de sua respectiva ordenação e o tempo de demora para ser atendido, ou seja, para iniciar o corte de cabelo.

No segundo programa, era necessário trocar os processos filhos por threads, mutex para exclusão mútua e semáforos ao invés de fila de mensagens. O número de cadeiras se mantém em sete, o de barbeiros aumenta para três e o de clientes para vinte e sete. A lógica e as impressões necessárias, funcionam de maneira similar ao primeiro programa.

Respostas das perguntas

Pergunta 1: Qual é o recurso comum que necessita de exclusão mútua ?

Resposta: O número de cadeiras disponíveis será um recurso comum do barbeiro e do cliente, portanto será necessário a exclusão mútua dessa variável.

Pergunta 2: De que maneira (leitura, escrita, ambos) barbeiros e clientes vão acessar o recurso comum ?

Resposta: No primeiro programa é necessário implementar um semáforo que trava o acesso ao recurso comum, que trava tanto o barbeiro quanto o cliente. No segundo programa isso é feito com o uso de mutex.

Pergunta 3: Como os números foram colocados na String ?

Resposta: Os números foram convertidos para string com o auxílio da função 'sprintf()'. A cada número convertido, era adicionado um espaço em branco, para facilitar a conversão de string para número novamente.

Pergunta 4: Como o barbeiro vai ter acesso aos valores a serem ordenados ?

Resposta: No primeiro programa, o cliente passa uma mensagem com a string para ser ordenada, é necessário converter em um vetor de inteiros e então ordenar em forma decrescente. No segundo programa, é implementado um vetor de struct, no qual o barbeiro acessa e pega a string do cliente.

Pergunta 5: Como o cliente vai ter acesso aos resultados?

Resposta: No primeiro programa, por meio de uma mensagem enviada pelo barbeiro. No segundo programa, o cliente acessa a struct após o barbeiro inserir as informações necessárias, para assim imprimir de maneira correta.

Resultados da execução do primeiro programa

```
cesarmanzano@cesarmanzano-Lenovo-ideapad-300-15ISK: ~/Downloads/Tarefa1
Arquivo Editar Ver Pesquisar Terminal Ajuda
=====
Cliente #10 foi atendido pelo barbeiro #1
Tempo aproximado para o cliente ser atendido: 0.01107000
String a ser ordenada:
819 376 51 943 568 280 765 545 613 363 5 733 888 923 332 157 981 858 304 303 145 247 53 582 244 776 495 606
289 770 971 31 123 1020 972 636 224 661 158 835 1022 107 491 833 7 821 989 932 602 216 159 692 462 156 197
650 931 636 233 197 383 127 226 451 125 122 10 293 781 166 105 726 272 594 482 223 338 394 79 885 609 236
500 1015 390 695 642 244 309 820 439 636 945 610 10 47 676 18 338 380 183 387 83 399 925 509 620
String ordenada:
1022 1020 1015 989 981 972 971 945 943 932 931 925 923 888 885 858 835 833 821 820 819 781 776 770 765 733
726 695 692 676 661 650 642 636 636 636 620 613 610 609 606 602 594 582 568 545 509 500 495 491 482 462 451
439 399 394 390 387 383 380 376 363 338 338 332 309 304 303 293 289 280 272 247 244 244 236 233 226 224 22
3 216 197 197 183 166 159 158 157 156 145 127 125 123 122 107 105 83 79 53 51 47 31 18 10 10 7 5
=====

=====
Cliente #19 foi atendido pelo barbeiro #2
Tempo aproximado para o cliente ser atendido: 0.01196500
String a ser ordenada:
615 228 971 579 282 756 941 852 1020 329 446 791 878 11 467 492 239 841 754 92 218 292 766 423 117 115 889
903 912 741 11 450 968 980 6 227 659 892 2 656 144 392 370 1020 401 836 436 585 600 113 621 816 349 364 216
411 477 28 291 366 768 246 814 659 203 765 830 806
String ordenada:
1020 1020 980 971 968 941 912 903 892 889 878 852 841 836 830 816 814 806 791 768 766 765 756 754 741 659 6
59 656 621 615 600 585 579 492 477 467 450 446 436 423 411 401 392 370 366 364 349 329 292 291 282 246 239
228 227 218 216 203 144 117 115 113 92 28 11 11 6 2
=====
```

Figura 1 - Resultado da execução do primeiro programa

```
cesarmanzano@cesarmanzano-Lenovo-ideapad-300-15ISK: ~/Downloads/Tarefa1
Arquivo Editar Ver Pesquisar Terminal Ajuda
=====
Cliente #2 foi atendido pelo barbeiro #2
Tempo aproximado para o cliente ser atendido: 0.01407200
String a ser ordenada:
261 147 105 385 537 298 711 840 377 512 503 1009 360 162 50 882 399 767 883 780 30 175 398 125 932 408 436
852 547 321 333 752 412 436 114 947 679 823 710 33 258 136 19 562 297 13 421 640 778 281 397 752 401 740 82
1 256 125 234 86 616 500 417 345 910 798 403 781 400 149 468 431 405 603 394 965 844 405 309 407 161 535 80
3 857 934 520 656 113 589 834 197 182 311 559 525 199 280 926 978 678 52 369 32 401 916 424 343 683 827 596
68 932 108 815 713 986 258 292 77 845 103 218 4 359 721 527 502 999 376 403 600 372 770 630 771 610 998 37
270 749 577 282 658 684 74 294 647 276 584 668 98 686 885 100 989 583 571 468 506 891 869 29 240 616 604 9
55 149 579 936 418 305 491 644 908 152 663 179 743 937 762 335 13 371 143 57 283 724 627 749 153 495 595 18
1 680 134 783 612 282 285 472 644 535 961 211 420 36 872 543 723 733 228 35 690 543 176 691 824 845 295 550
996 735 68 100 338 147 827 894 373 90 343 1015 623 281 149 987 315 1020 453 16 676 680 1016 343 146 170 11
915
String ordenada:
1020 1016 1015 1009 999 998 996 989 987 986 978 965 961 955 947 937 936 934 932 932 926 916 915 910 908 894
891 885 883 882 872 869 857 852 845 845 844 840 834 827 827 824 823 821 815 803 798 783 781 780 778 771 77
0 767 762 752 752 749 749 743 740 735 733 724 723 721 713 711 710 691 690 686 684 683 680 680 679 678 676 6
68 663 658 656 647 644 644 640 630 627 623 616 616 612 610 604 603 600 596 595 589 584 583 579 577 571 562
559 550 547 543 543 537 535 535 527 525 520 512 506 503 502 500 495 491 472 468 468 453 436 436 431 424 421
420 418 417 412 408 407 405 405 403 403 401 401 400 399 398 397 394 385 377 376 373 372 371 369 360 359 34
5 343 343 343 338 335 333 321 315 311 309 305 298 297 295 294 292 285 283 282 282 281 281 280 276 270 261 2
58 258 256 240 234 228 218 211 199 197 182 181 179 176 175 170 162 161 153 152 149 149 149 147 147 146 143
136 134 125 125 114 113 108 105 103 100 100 98 90 86 77 74 68 68 57 52 50 37 36 35 33 32 30 29 19 16 13 13
11 4
=====
```

Figura 2 - Resultado da execução do primeiro programa

```
cesarmanzano@cesarmanzano-Lenovo-ideapad-300-15ISK: ~/Downloads/Tarefa1
Arquivo Editar Ver Pesquisar Terminal Ajuda

=====
Cliente #17 foi atendido pelo barbeiro #1
Tempo aproximado para o cliente ser atendido: 0.00885800
String a ser ordenada:
981 810 47 30 593 939 809 299 88 352 368 941 424 846 645 989 835 812 524 517 1014 794 891 854 819 714 755 6
85 232 437 759 191 224 804 219 762 667 973 38 699 248 351 618 671 120 240 583 953 997 30 393 934 823 261 76
5 565 919 498 173 129 879 877 264 27 658 427 733 248 323 715 946 570 43 487 164 161 725 691 38 645 719 375
556 465 581 245 8 477 687 179 604 489 33 866 514 636 271 224 828 538 938 751 85 925 215 193 64 885 882 46 4
53 579 419 1008 21 998 230 994 453 861 151 1001 327 128 845 840 708 39 1008 514 575 923 188 605 826 402 796
834 210 656 878 661 158 220 646 177 196 820 95 593 658 244 571 984 316 339 747 2 376 732 460 896 579 646 4
24 328 992 197 139 179 797 961 785 953 158 354 54 298 152 147 889 754 335 384 661 649 721 331 595 21 41 999
915 564 623 316 890 592 457 973 716 232 857 478 108 959 830 160 235 926 251 47 658 584 429 242 157 128 572
696 147 611 673 1006
String ordenada:
1014 1008 1008 1006 1001 999 998 997 994 992 989 984 981 973 973 961 959 953 953 946 941 939 938 934 926 92
5 923 919 915 896 891 890 889 885 882 879 878 877 866 861 857 854 846 845 840 835 834 830 828 826 823 820 8
19 812 810 809 804 797 796 794 785 765 762 759 755 754 751 747 733 732 725 721 719 716 715 714 708 699 696
691 687 685 673 671 667 661 661 658 658 658 656 649 646 646 645 645 636 623 618 611 605 604 595 593 593 592
584 583 581 579 579 575 572 571 570 565 564 556 538 524 517 514 514 498 489 487 478 477 465 460 457 453 45
3 437 429 427 424 424 419 402 393 384 376 375 368 354 352 351 339 335 331 328 327 323 316 316 299 298 271 2
64 261 251 248 248 245 244 242 240 235 232 232 230 224 224 220 219 215 210 197 196 193 191 188 179 179 177
173 164 161 160 158 158 157 152 151 147 147 139 129 128 128 120 108 95 88 85 64 54 47 47 46 43 41 39 38 38
33 30 30 27 21 21 8 2
=====
```

Figura 3 - Resultado da execução do primeiro programa

```
cesarmanzano@cesarmanzano-Lenovo-ideapad-300-15ISK: ~/Downloads/Tarefa1
Arquivo Editar Ver Pesquisar Terminal Ajuda

3 582 582 579 570 565 564 562 561 557 553 549 541 540 540 539 536 535 522 516 498 494 488 483 470 467 460 4
59 456 455 454 453 448 437 433 431 427 422 419 415 414 413 406 396 396 396 389 389 387 383 376 374 372 372
361 360 341 341 331 328 323 318 317 312 308 302 294 289 287 281 277 275 266 251 240 226 217 216 212 210 208
196 196 195 189 187 185 181 172 171 170 170 168 164 164 163 163 156 155 152 151 150 143 136 134 132 130 12
7 122 118 110 107 99 97 96 95 92 91 88 87 87 81 67 56 52 46 41 27 26 26 24 21 11 11 6
=====

=====
Cliente #14 foi atendido pelo barbeiro #1
Tempo aproximado para o cliente ser atendido: 0.00702600
String a ser ordenada:
948 403 848 548 984 300 61 634 864 556 835 55 501 485 470 870 765 292 971 259 248 861 920 865 299 388 746 6
95 886 484 433 811 831 204 283 792 502 342 403 289 843 161 342 321 591 757 169 333 993 63 590 164 868 433 6
144 766 696 784 629 103
String ordenada:
993 984 971 948 920 886 870 868 865 864 861 848 843 835 831 811 792 784 766 765 757 746 696 695 634 629 591
590 556 548 502 501 485 484 470 433 433 403 403 388 342 342 333 321 300 299 292 289 283 259 248 204 169 16
4 161 144 103 63 61 55 6
=====

=====
Cliente #16 foi atendido pelo barbeiro #1
Tempo aproximado para o cliente ser atendido: 0.00827900
String a ser ordenada:
138 278 474 999 162 653 631 527 80 800 891 269 305 369 176 590 988 725 85 766 842 567 1021 416 245 847 536
```

Figura 4 - Resultado da execução do primeiro programa

```
cesarmanzano@cesarmanzano-Lenovo-ideapad-300-15ISK: ~/Downloads/Tarefa1
Arquivo Editar Ver Pesquisar Terminal Ajuda
88 464 460 459 438 430 419 416 404 400 396 396 392 390 384 382 376 372 370 368 361 342 342 337 334 334 333
332 331 323 321 318 318 309 308 303 292 291 289 282 279 274 272 257 249 241 224 224 221 217 217 216 215 212
185 185 178 172 156 148 147 146 140 140 133 128 127 125 108 104 94 89 86 86 80 76 75 71 64 64 62 55 51 48
41 40 37 33 31 23 16 11 11 4 2
=====
Cliente #12 nao foi atendido
Cliente #11 nao foi atendido
Cliente #7 nao foi atendido
=====
Cliente #5 foi atendido pelo barbeiro #1
Tempo aproximado para o cliente ser atendido: 0.00507100
String a ser ordenada:
334 711 676 362 711 977 601 252 193 879 754 994 853 988 538 400 73 684 92 524 1018 453 437 167 50 577 142 4
4 959 39 402 271 748 56 577 383 977 155 633 93 979 364 64 809 275 600 186 347 208 222 815 203 619 230 368 6
13 751 454 602 687 491 1002 902 163 1002 457 544 957
String ordenada:
1018 1002 1002 994 988 979 977 977 959 957 902 879 853 815 809 754 751 748 711 711 687 684 676 633 619 613
602 601 600 577 577 544 538 524 491 457 454 453 437 402 400 383 368 364 362 347 334 275 271 252 230 222 208
203 193 186 167 163 155 142 93 92 73 64 56 50 44 39
=====
Cliente #15 nao foi atendido
=====
Cliente #20 foi atendido pelo barbeiro #1
```

Figura 5 - Resultado da execução do primeiro programa

```
cesarmanzano@cesarmanzano-Lenovo-ideapad-300-15ISK: ~/Downloads/Tarefa1
Arquivo Editar Ver Pesquisar Terminal Ajuda
709 708 704 701 698 696 695 694 677 676 670 660 638 636 624 620 617 614 613 609 600 599 596 595 592 591 590
588 587 583 579 574 574 573 572 569 567 567 567 563 558 548 544 543 542 539 539 532 527 524 520 520 519 51
9 518 514 512 509 505 503 494 489 476 473 471 471 468 464 449 449 443 435 434 433 424 422 422 409 407 405 4
02 400 399 398 396 390 390 385 385 384 377 372 372 368 356 353 352 350 349 349 348 344 342 340 331 331 330
329 318 315 314 300 284 282 282 279 278 277 275 274 269 262 258 256 251 247 242 241 237 228 223 211 201 198
193 190 189 184 177 176 171 164 160 148 142 136 135 131 129 128 127 123 120 119 113 112 111 105 97 97 86 8
5 83 81 76 72 71 65 64 60 57 53 42 41 40 28 17 14 10 9 7 6 5 5
=====
=====
Cliente #7 foi atendido pelo barbeiro #2
Tempo aproximado para o cliente ser atendido: 0.01308400
String a ser ordenada:
645 230 275 801 479 456 425 202 731 125 644 925 417 726 372 300 349 434 826 624 249 266
String ordenada:
925 826 801 731 726 645 644 624 479 456 434 425 417 372 349 300 275 266 249 230 202 125
=====
=====
Cliente #5 foi atendido pelo barbeiro #1
Tempo aproximado para o cliente ser atendido: 0.01518600
String a ser ordenada:
490 911 1022 498 853 1009 189 990 959 297 983 102 625 95 514 903 674 846 433 318 983 327 483 315 794 836 21
8 450 795 735 1017 262 623 962 704 399 895 891 312 831 165 272 931 788 311 422 614 930 191 24 225 97 295 70
6 357 67 519 573 515 237 285 455 497 852 395 178 228 267 1013 484 21 155 701 896 888 1010 296 479 917 431 4
```

Figura 6 - Resultado da execução do primeiro programa

```
cesarmanzano@cesarmanzano-Lenovo-ideapad-300-15ISK: ~/Downloads/Tarefa1
Arquivo Editar Ver Pesquisar Terminal Ajuda
String ordenada:
839 788 506 487 487 325 93
=====

=====
Cliente #2 foi atendido pelo barbeiro #2
Tempo aproximado para o cliente ser atendido: 0.00150900
String a ser ordenada:
365 486 732 580 146 375 953 655 56 261 14 804 900 176 167 987 85 949 801 307 725 929 784 823 427 677 881 12
220 390 912 583 821 622 87 965 941 984 597 995 223 556 722 46 730 833 978 759 706 756 44 354 662 826 154 1
2 427 979 22 645 346 878 151 144 477 182 32 395 144 574 313 311 107 13 355 781 790 310 518 473 1010 506 771
595 255 923 551 626 879 571 248 203 427 344 291 848 524 322 221 612 894 532 921 945 489 254
String ordenada:
1010 995 987 984 979 978 965 953 949 945 941 929 923 921 912 900 894 881 879 878 848 833 826 823 821 804 80
1 790 784 781 771 759 756 732 730 725 722 706 677 662 655 645 626 622 612 597 595 583 580 574 571 556 551 5
32 524 518 506 489 486 477 473 427 427 427 395 390 375 365 355 354 346 344 322 313 311 310 307 291 261 255
254 248 223 221 220 203 182 176 167 154 151 146 144 144 107 87 85 56 46 44 32 22 14 13 12 12
=====

Cliente #18 nao foi atendido

=====
Cliente #3 foi atendido pelo barbeiro #1
Tempo aproximado para o cliente ser atendido: 0.00454100
String a ser ordenada:
496 976 644 965 254 353 423 611 3 849 832 672 913 657 98 413 681 276 604 384 208 550 931 21 520 834 139 175
```

Figura 7 - Resultado da execução do primeiro programa

Resultados da execução do segundo programa

```
=====
Cliente #12 atendido pelo barbeiro #2
Tempo de atendimento: 0.01005700

String não ordenada: 991 389 775 18 630 874 251 714 639 757 384 397 552 797 843
921 471 262 209 740 408 56 640 346 192

String ordenada: 18 56 192 209 251 262 346 384 389 397 408 471 552 630 639 640
714 740 757 775 797 843 874 921 991
```

Figura 8 - Resultado da execução do segundo programa

```
=====
Cliente #3 atendido pelo barbeiro #2
Tempo de atendimento: 0.01021200

String não ordenada: 398 119 752 402 113 985 585 285 438 377 784 80 543 365 359
518 864 468 347 312 308 905 422 35

String ordenada: 35 80 113 119 285 308 312 347 359 365 377 398 402 422 438 468
518 543 585 752 784 864 905 985
```

Figura 9 - Resultado da execução do segundo programa

```
=====
Cliente #13 atendido pelo barbeiro #2
Tempo de atendimento: 0.01018600

String não ordenada: 652 421 129 165 122 318 516 345 373 918 238 383 849 952 99
5 158 902 979 892 895 24 1018 850 336 137 397 409 210 628 993 412 422 456 655 8
59 821 698 514 806 887 460 867 340 823 535 886 661 313 135 607 989 152 1005 988
71 900 384 626 178 896 239 671 866 597 658 1011 498 234 834 306 646 30 79 921
126 729 266 812 299 1021 780 591 88 720 1003

String ordenada: 24 30 71 79 88 122 126 129 135 137 152 158 165 178 210 234 238
239 266 299 306 313 318 336 340 345 373 383 384 397 409 412 421 422 456 460 49
8 514 516 535 591 597 607 626 628 646 652 655 658 661 671 698 720 729 780 806 8
12 821 823 834 849 850 859 866 867 886 887 892 895 896 900 902 918 921 952 979
988 989 993 995 1003 1005 1011 1018 1021
```

Figura 10 - Resultado da execução do segundo programa

```
=====
Cliente #23 atendido pelo barbeiro #2
Tempo de atendimento: 0.01017500

String não ordenada: 906 723 527 1002 184 725 500 405 255 438 713 739 81 518 66
3 819 887 522 416 457 761 57 257 637 382 316 30 31 930 478 373 980 417 221 880
954 798 204 521 614 728 9 138 630 268 294 510 6 835 720 333 205 894 20 710 825
783 927 246 31 594 948 712

String ordenada: 6 9 20 30 31 31 57 81 138 184 204 205 221 246 255 257 268 294
316 333 373 382 405 416 417 438 457 478 500 510 518 521 522 527 594 614 630 637
663 710 712 713 720 723 725 728 739 761 783 798 819 825 835 880 887 894 906 92
7 930 948 954 980 1002
```

Figura 11 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #1 atendido pelo barbeiro #2

Tempo de atendimento: 0.01036600

String não ordenada: 354 763 336 850 444 846 941 694 524 36 761 513 136 957 907
449 693 975 528 129 383 944 210 478 441 206 586 370 799 465 610 783 864 954 19
235 847 34 270 329 825 690 402 660 437 412 630 908 537 975 616 537 876 761 771
42 459 646 300 497 415 585

String ordenada: 19 34 36 42 129 136 206 210 235 270 300 329 336 354 370 383 40
2 412 415 437 441 444 449 459 465 478 497 513 524 528 537 537 585 586 610 616 6
30 646 660 690 693 694 761 761 763 771 783 799 825 846 847 850 864 876 907 908
941 944 954 957 975 975
```

Figura 12 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #8 atendido pelo barbeiro #2

Tempo de atendimento: 0.01048600

String não ordenada: 41 270 414 257 628 678 577 852 379 674 27

String ordenada: 27 41 257 270 379 414 577 628 674 678 852
```

Figura 13 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #5 atendido pelo barbeiro #3

Tempo de atendimento: 0.01053800

String não ordenada: 873 689 906 256 598 62 959 269 568 445 989 660 1008 663 44
2 299 900 466 639 125 429 439 182 881 531 594 510 32 52 1016 866 38 899 427 863
26 958 4 479 252 775 851 625 217 280 477

String ordenada: 4 26 32 38 52 62 125 182 217 252 256 269 280 299 427 429 439 4
42 445 466 477 479 510 531 568 594 598 625 639 660 663 689 775 851 863 866 873
881 899 900 906 958 959 989 1008 1016
```

Figura 14 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #4 atendido pelo barbeiro #1

Tempo de atendimento: 0.01058100

String não ordenada: 191 657 584 56 32 618 641 638 887 231 61 621 764 508 290 4
1 522 770 9 991 813 551 993 891 586 893 654 833 964 263 615 644 626 447 982 757
176 130 982 263 627 984 892 731 417 223 959 630 317 677 20 845 24 33 543 495 9
03 453 341 494 343 451 218 164 461 778 905 839 210 78 700 9 61 245 33 976 605 2
31 842 512 223 915 829 987 1018

String ordenada: 9 9 20 24 32 33 33 41 56 61 61 78 130 164 176 191 210 218 223
223 231 231 245 263 263 290 317 341 343 417 447 451 453 461 494 495 508 512 522
543 551 584 586 605 615 618 621 626 627 630 638 641 644 654 657 677 700 731 75
7 764 770 778 813 829 833 839 842 845 887 891 892 893 903 905 915 959 964 976 9
82 982 984 987 991 993 1018
```

Figura 15 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24

Cliente #11 atendido pelo barbeiro #1

Tempo de atendimento: 0.01058200

String não ordenada: 177 872 943 311 658 220 195 414 243 903 86 824 384 601 912
377 813 981 600 61 977 410 69 564 572 281 740 620 902 517 50 195 105 190 781 4
62 285 260 344 668 1004 773 310 601 36 919 162 531 100 662 798 889 519 913 803
867 38 363 377 773 376 418 85 127 841 599 582 560 568 913 23 43 410 106 844 390
91 913 503 372 487 855 429 904 235 83 350 245 777 189 16 366 967 390 427 141 1
36 665 934

String ordenada: 16 23 36 38 43 50 61 69 83 85 86 91 100 105 106 127 136 141 16
2 177 189 190 195 195 220 235 243 245 260 281 285 310 311 344 350 363 366 372 3
76 377 377 384 390 390 410 410 414 418 427 429 462 487 503 517 519 531 560 564
568 572 582 599 600 601 601 620 658 662 665 668 740 773 773 777 781 798 803 813
824 841 844 855 867 872 889 902 903 904 912 913 913 913 919 934 943 967 977 98
1 1004
```

Figura 16 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24

Cliente #26 atendido pelo barbeiro #3

Tempo de atendimento: 0.01061700

String não ordenada: 21 304 88 1003 304 88 1003 214 267 118 20 119 667 772 284
790 971 898 858 333 216 370 425 498 553 227 452 913 133 629 809 818 739 986 601
20 666 839 635 972 279 427 822 332 991 7 338 630 515

String ordenada: 7 20 20 21 88 88 118 119 133 214 216 227 267 279 284 304 304 3
32 333 338 370 425 427 452 498 515 553 601 629 630 635 666 667 739 772 790 809
818 822 839 858 898 913 971 972 986 991 1003 1003
```

Figura 17 - Resultado da execução do segundo programa

```
Cliente #6 atendido pelo barbeiro #1

Tempo de atendimento: 0.01079100

String não ordenada: 587 721 260 403 143 580 203 921 249 606 843 646 231 818 59
4 557 310 215 301 227 98 380 392 924 476 348 419 252 214 748 96 453 98 407 744
369 719 899 997 294 870 665 411 724 196 731 748 174 771 42 589 889 64 913 994 3
41 886 776 343 211 225 790 162 516 618 379 784 49 147 423 797 102 788 69 563 98
387 838 888 181 994 153 1017 156 493 60

String ordenada: 42 49 60 64 69 96 98 98 98 102 143 147 153 156 162 174 181 196
203 211 214 215 225 227 231 249 252 260 294 301 310 341 343 348 369 379 380 38
7 392 403 407 411 419 423 453 476 493 516 557 563 580 587 589 594 606 618 646 6
65 719 721 724 731 744 748 748 771 776 784 788 790 797 818 838 843 870 886 888
889 899 913 921 924 994 994 997 1017
```

Figura 18 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24

Cliente #17 atendido pelo barbeiro #3

Tempo de atendimento: 0.01075900

String não ordenada: 502 34 12 366 718 378 452 612 886 287 151 988 781 592 361
402 726 422 508 965 451 434 9 194 666

String ordenada: 9 12 34 151 194 287 361 366 378 402 422 434 451 452 502 508 59
2 612 666 718 726 781 886 965 988
```

Figura 19 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #24 atendido pelo barbeiro #2
Tempo de atendimento: 0.01073400
String não ordenada: 567 755 902 128 750 169 765 36 1010 599 567 725
String ordenada: 36 128 169 567 567 599 725 750 755 765 902 1010
```

Figura 20 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #25 atendido pelo barbeiro #3
Tempo de atendimento: 0.01074900
String não ordenada: 281 787 256 328 242 634 9 688 691 813 474 764 622 828 967
261 728 935 708 714 46 960 677 670 272 792 816 418 233 963 822 789 910 234 642
566 320 973 536 698
String ordenada: 9 46 233 234 242 256 261 272 281 320 328 418 474 536 566 622 6
34 642 670 677 688 691 698 708 714 728 764 787 789 792 813 816 822 828 910 935
960 963 967 973
```

Figura 21 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #21 atendido pelo barbeiro #1
Tempo de atendimento: 0.01091600
String não ordenada: 431 153 373 127 720 627 126 474 125 370
String ordenada: 125 126 127 153 370 373 431 474 627 720
```

Figura 22 - Resultado da execução do segundo programa

```
Cliente #7 atendido pelo barbeiro #1
Tempo de atendimento: 0.01101500
String não ordenada: 327 238 92 110 62 160 259 200 698
String ordenada: 62 92 110 160 200 238 259 327 698
```

Figura 23 - Resultado da execução do segundo programa

```
Cliente #18 atendido pelo barbeiro #3
Tempo de atendimento: 0.01097600
String não ordenada: 242 572 865 73 690 11 508 912 314 194 502 561 949 735 292
999 384 724 117 676 903 27 580 997 286 330 730 631 295 762 366 164 463
String ordenada: 11 27 73 117 164 194 242 286 292 295 314 330 366 384 463 502 5
08 561 572 580 631 676 690 724 730 735 762 865 903 912 949 997 999
```

Figura 24 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #22 atendido pelo barbeiro #3

Tempo de atendimento: 0.01098600

String não ordenada: 92 185 695 274 212 71 391 105 880 584 415 145 934 375 732
222 261 167 754 746 309 464 707 855 762 922 217 661 292 755 209 200 286 810 683
885 146 845 630 490 293 750 862 915 653 528 416 1017 974 484 937 480 965 922 6
20 231 939 852 15 277 188 891 440 977 855 276 91 703 437 481 592 51 97 576 166
482 948 372 724 509 369 267 421 506 988 137 58 740 60 10 240 739 429 826 436

String ordenada: 10 15 51 58 60 71 91 92 97 105 137 145 146 166 167 185 188 200
209 212 217 222 231 240 261 267 274 276 277 286 292 293 309 369 372 375 391 41
5 416 421 429 436 437 440 464 480 481 482 484 490 506 509 528 576 584 592 620 6
30 653 661 683 695 703 707 724 732 739 740 746 750 754 755 762 810 826 845 852
855 855 862 880 885 891 915 922 922 934 937 939 948 965 974 977 988 1017
```

Figura 25 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #14 atendido pelo barbeiro #3

Tempo de atendimento: 0.01106900

String não ordenada: 366 453 504 365 688 836 834 997 54 58 145 422 72 86 126 36
3 259 675 554 997 714 959 39 379 135 151 318 430 790 778 663 837 729 635 687 14
3 406 388 303 876 555 734 126 309 398 119 45 85 362 263 763 658 541 891 323 286
939 814 663 244 717 867 301 236 226 301 411 376 776 989 668

String ordenada: 39 45 54 58 72 85 86 119 126 126 135 143 145 151 226 236 244 2
59 263 286 301 301 303 309 318 323 362 363 365 366 376 379 388 398 406 411 422
430 453 504 541 554 555 635 658 663 663 668 675 687 688 714 717 729 734 763 776
778 790 814 834 836 837 867 876 891 939 959 989 997 997
```

Figura 26 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #27 atendido pelo barbeiro #3

Tempo de atendimento: 0.01100700

String não ordenada: 756 336 410 182 780 785 383 672 821 828 626 323 13 105 29
42 764 933 32 580 167 494 384 495 858 908 538 8 980 471 163 963 293 646 667 978
733 153 945

String ordenada: 8 13 29 32 42 105 153 163 167 182 293 323 336 383 384 410 471
494 495 538 580 626 646 667 672 733 756 764 780 785 821 828 858 908 933 945 963
978 980
```

Figura 27 - Resultado da execução do segundo programa

```
Experimentos-SO-A-Exp5 — zsh — 79x24
Cliente #19 atendido pelo barbeiro #3

Tempo de atendimento: 0.01110600

String não ordenada: 977 604 166 273 182 476 773 543 1016 355 356 547 225 890 4
44 236 815 473 800 831 572 989 737 19 231 84 639 851 404 494 617 579 683 418 72
4 183 993 717 671 282 689 789 25 186 291 307 480 442 712 883 972 711 996 944 13
2 812 91 454 750 859 868 442 224 868 449 521 209 887 229 405 968 44 91 386 135
582 345

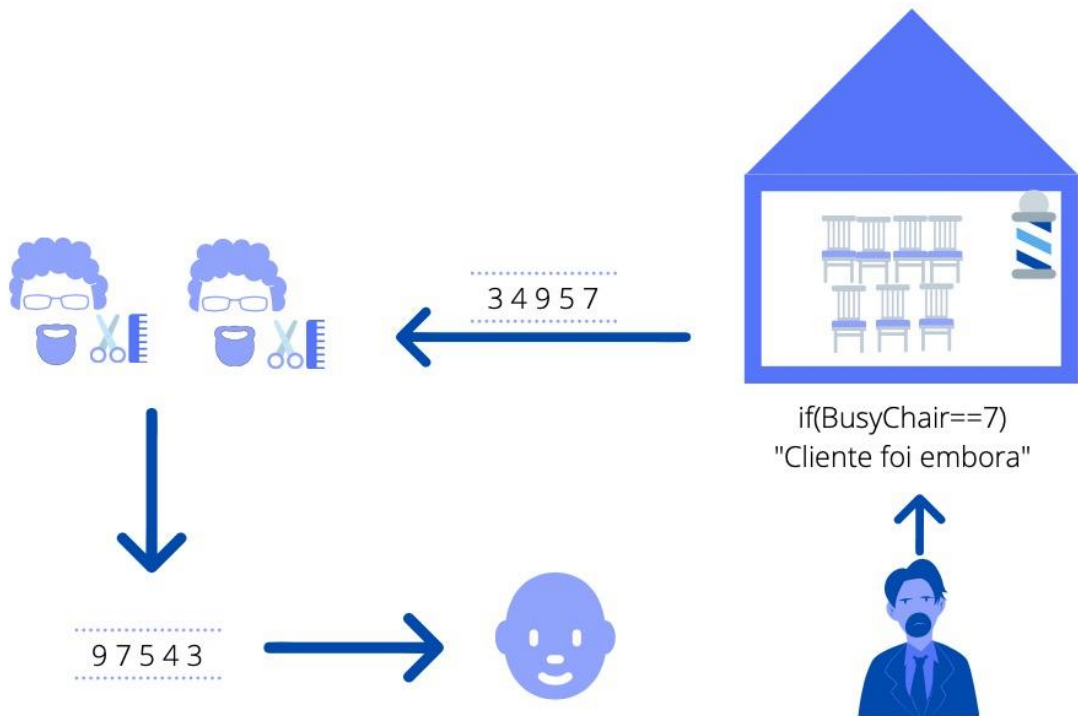
String ordenada: 19 25 44 84 91 91 132 135 166 182 183 186 209 224 225 229 231
236 273 282 291 307 345 355 356 386 404 405 418 442 442 444 449 454 473 476 480
494 521 543 547 572 579 582 604 617 639 671 683 689 711 712 717 724 737 750 77
3 789 800 812 815 831 851 859 868 868 883 887 890 944 968 972 977 989 993 996 1
016
```

Figura 28 - Resultado da execução do segundo programa

Análise dos resultados

Programa 1

Inicialmente mostraremos dois esquemas para exemplificar a implementação do primeiro programa do barbeiro dorminhoco.



Neste programa foram usados quatro mecanismos vistos ao longo do semestre, que estão listados abaixo:

- Fila de mensagens;
- Memória compartilhada;
- Semáforos;
- Criação de processos filhos.

São criados vinte processos clientes e dois processos barbeiros. Os processos clientes verificavam se havia alguma cadeira de espera disponível e caso não houvesse, o cliente saía da barbearia. Caso contrário, o processo gerava um vetor de inteiros com números aleatórios entre 2 e 1023, com tamanho máximo de 300 (a seed era diferente para cada processo, gerando valores diferentes para cada cliente). Após isso era necessário converter esse vetor de inteiros em um vetor de caracteres, ou seja, uma string, que seria enviada ao barbeiro. O barbeiro por sua vez, pegava a mensagem enviada pelo cliente, convertia a string para um vetor de inteiros novamente, para assim poder ser ordenado de forma decrescente. Para a ordenação foi escolhido o método bubblesort, uma vez que a quantidade de elementos a serem ordenados não era alta, causando um baixo impacto no tempo de execução do programa. Após a ordenação era necessário converter o vetor de inteiros para string novamente, para poder enviar outra mensagem ao cliente. O cliente por sua vez, ao receber uma mensagem, imprimia as informações necessárias: número do barbeiro que o atendeu, seu próprio número, a string a ser ordenada, a string ordenada e o tempo de atendimento. Para termos certeza de que cada barbeiro devolveria a mensagem ao cliente certo, utilizou-se dos tipos de mensagens da fila. O tipo de mensagem de para o envio ao barbeiro era o tipo 1 e do envio do barbeiro para o cliente utiliza-se o número do cliente mais 50 (numero do cliente + 50) com tipo de mensagem (a informação do número do cliente era passada na struct de envio). Dessa forma garantíamos que cada barbeiro enviaria a mensagem para o cliente correto.

A respeito dos resultados, podemos observar que esta parte do experimento foi realizada com sucesso. Verifica-se que a string de cada processo filho foi ordenada de forma decrescente, como esperado e que alguns clientes também não foram atendidos.

Para uma melhor análise discussão do programa, foram feitas 4 rodadas de testes (execução) e um gráfico para cada uma dessas rodadas, assim podendo observar melhor alguns aspectos do programa. Os gráficos são mostrados a seguir:



Tabela 1 - Tempo de atendimento por cliente na primeira execução

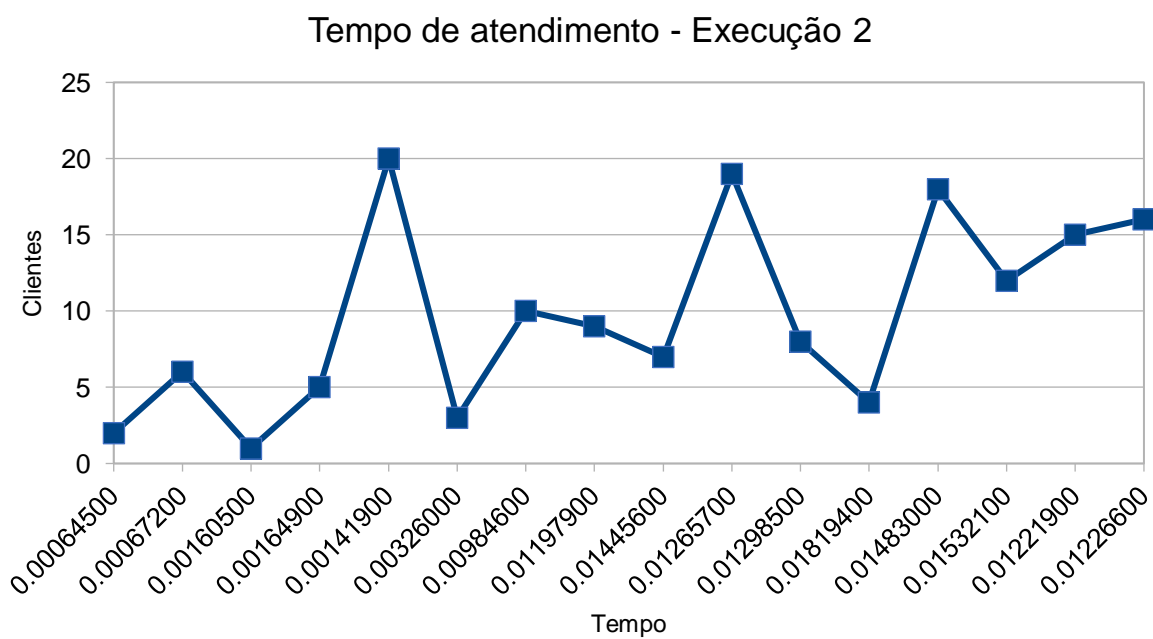


Tabela 2 - Tempo de atendimento por cliente na segunda execução

Tempo de Atendimento - Execução 3

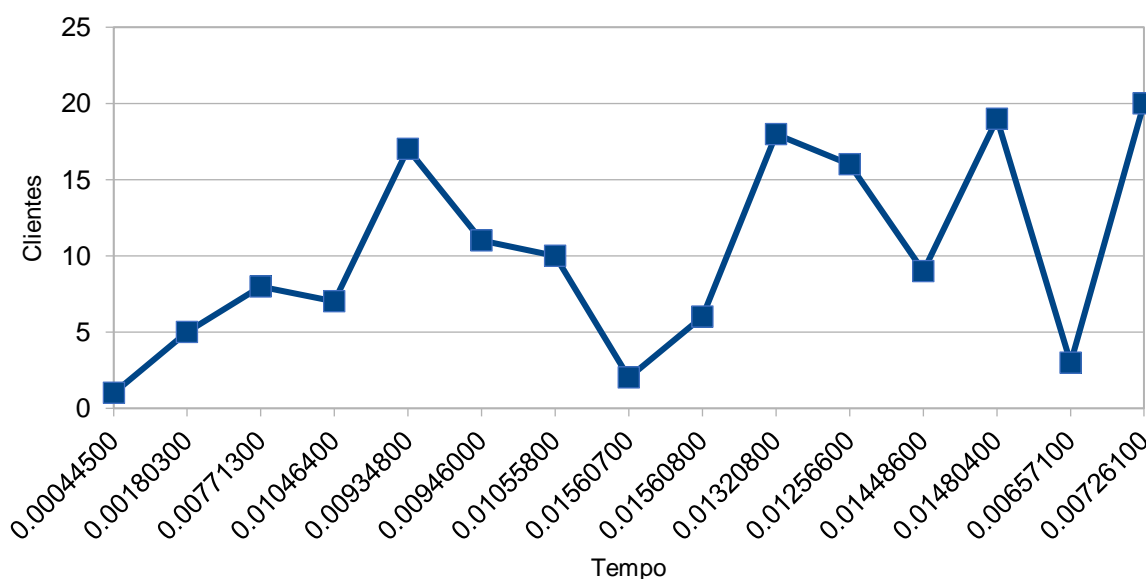


Tabela 3 - Tempo de atendimento por cliente na terceira execução

Tempo de atendimento - Execução 4

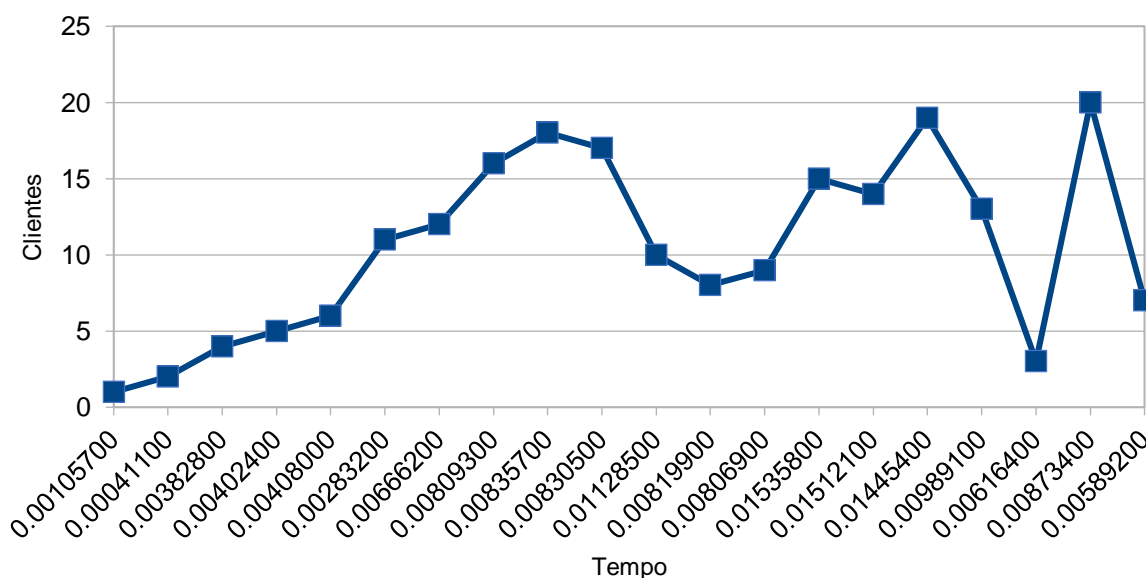


Tabela 4 - Tempo de atendimento por cliente na quarta execução

Nos gráficos acima podemos ver que a ordem dos clientes não afetou o seu tempo, ou seja, os primeiros clientes não necessariamente foram os primeiros a serem atendidos. Isso se deve ao fato de não sabermos como o SO escalona os processos. Os primeiros processos escalonados pelo SO serão os primeiros a entrarem na fila de mensagens e, portanto, os primeiros a serem escalonados. Alguns outros fatores como programas externos rodando em background, podem interferir nos resultados, atrasando ou adiantando o atendimento dos clientes.

Também podemos verificar nos gráficos que nem todos os vinte clientes foram atendidos, como mostrado na análise abaixo:



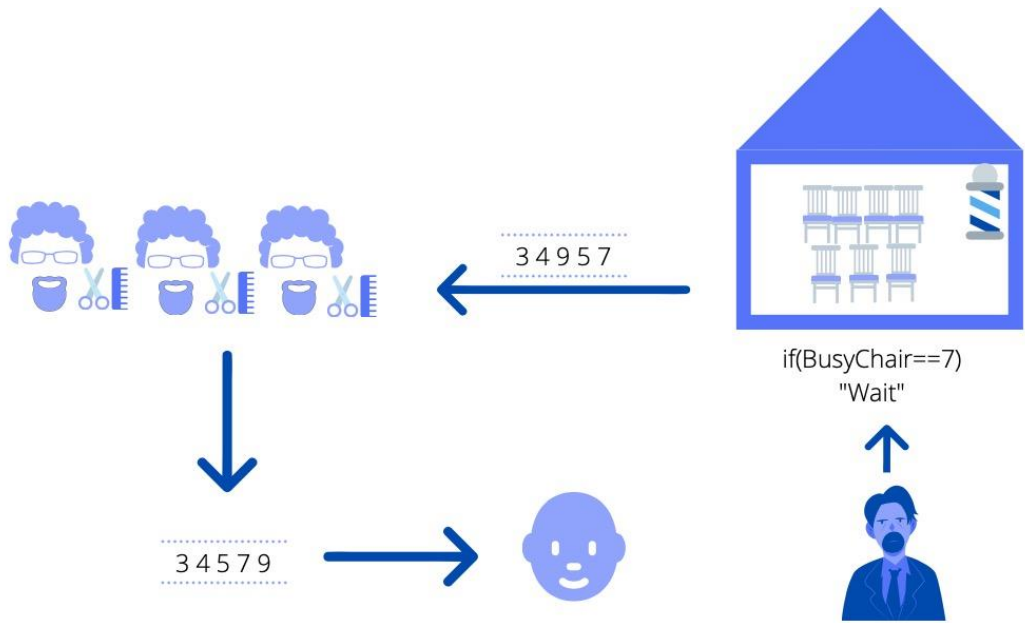
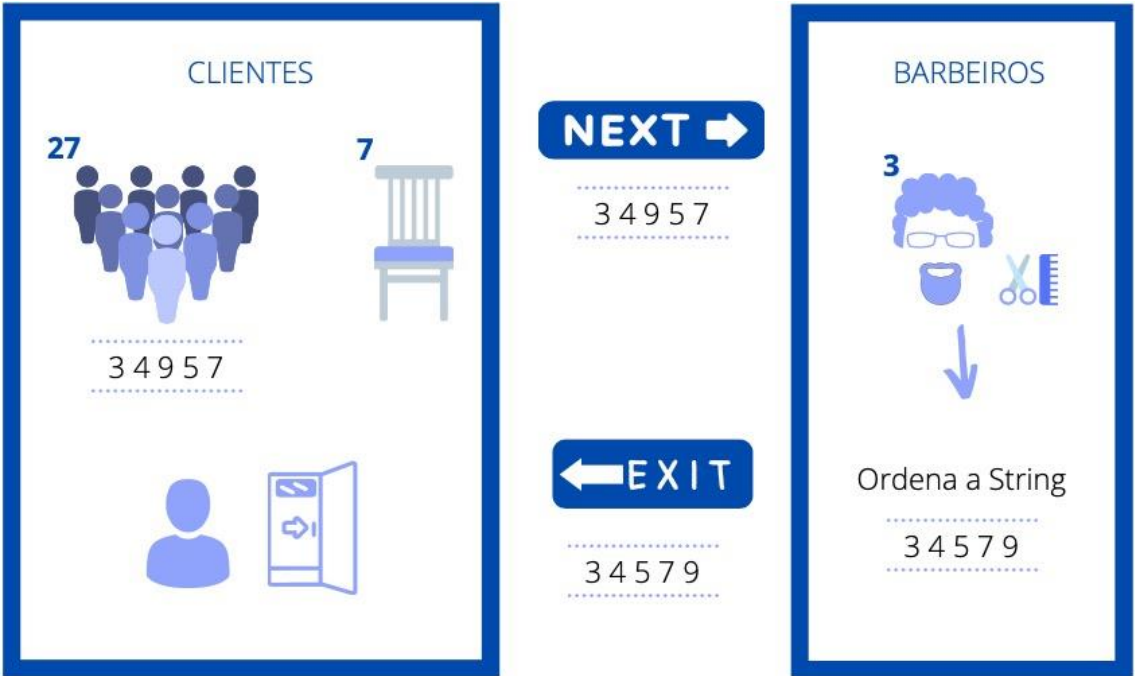
Tabela 5 - Números de clientes não atendidos a cada rodada de execução

Neste gráfico podemos verificar a quantidade de clientes que não foram atendidos a cada execução. Na média foram 2,75 clientes não atendidos por execução. Isso mostra que em alguns casos, alguns clientes (processos) não são terminados em tempo suficiente para outros serem atendidos.

Um único processo cliente precisa enviar uma mensagem ao barbeiro que, por sua vez, ordenará a string e enviará outra mensagem ao cliente e este imprime as informações necessárias. Portanto, a quantidade de tarefas que um único processo precisa realizar é muito alta e por isso, alguns clientes não são atendidos.

Programa 2

Inicialmente mostraremos dois esquemas para exemplificar a implementação do segundo programa do barbeiro dorminhoco



Nessa segunda implementação do programa do barbeiro dorminhoco, o número total de cadeiras se manteve igual, o de barbeiros aumentou para três e o de clientes para vinte e sete. Nesta implementação do programa, todos os clientes deveriam ser atendidos. Em relação aos mecanismos usados, também houve algumas diferenças significativas:

- Threads ao invés de processos filhos;
- Semáforo no lugar de fila de mensagens;
- Mutex para exclusão mútua.

O grande problema a ser resolvido nesse programa, era a comunicação efetiva entre as threads, ou seja, o barbeiro necessitava manipular a string de um cliente e devolver o resultado para este mesmo cliente. Como o uso de fila de mensagens não poderia ser feito, era necessário o uso de semáforos e mutex, para que isto ocorra.

Inicialmente era necessário criar um vetor de struct, contendo as seguintes informações: número do barbeiro, número do cliente atual, string ordenada e não ordenada e o tamanho da string. O cliente inicia gerando um vetor de inteiros que é convertido em string e armazenado no vetor de struct, na posição correspondente ao seu número. O cliente também coloca no campo de 'número de cliente atual' o valor zero, demonstrando que está pronto para cortar o cabelo, podendo ser interpretado como uma mensagem ao barbeiro. O barbeiro irá percorrer o vetor de struct verificando se há alguma 'mensagem' do cliente. Caso ache esta 'mensagem', colocará o valor 1 no campo correspondente, ou seja, a mensagem foi recebida. Após isso o barbeiro manipula a string, ou seja, ordena de forma crescente, e coloca essa string na posição correta do cliente. Após a ordenação da string, o cliente está liberado para exibir as informações necessárias: tempo de atendimento, seu número e do barbeiro que o atendeu, string antes e depois da ordenação.

Vale ressaltar que o método de ordenação escolhido foi novamente o bubblesort, cuja complexidade se dá por $O(n^2)$, uma vez que a quantidade de números a serem ordenados, não demandava um algoritmo com complexidade melhor.

A respeito dos resultados, podemos observar que esta parte do experimento foi realizada com sucesso. Verifica-se que a string de cada thread cliente foi ordenada corretamente, em ordem crescente, pelo barbeiro.

Para uma melhor análise discussão do programa, foram feitas 5 rodadas de testes (execução) e um gráfico para cada uma dessas rodadas, assim podendo observar melhor alguns aspectos do programa. Os gráficos são mostrados a seguir:

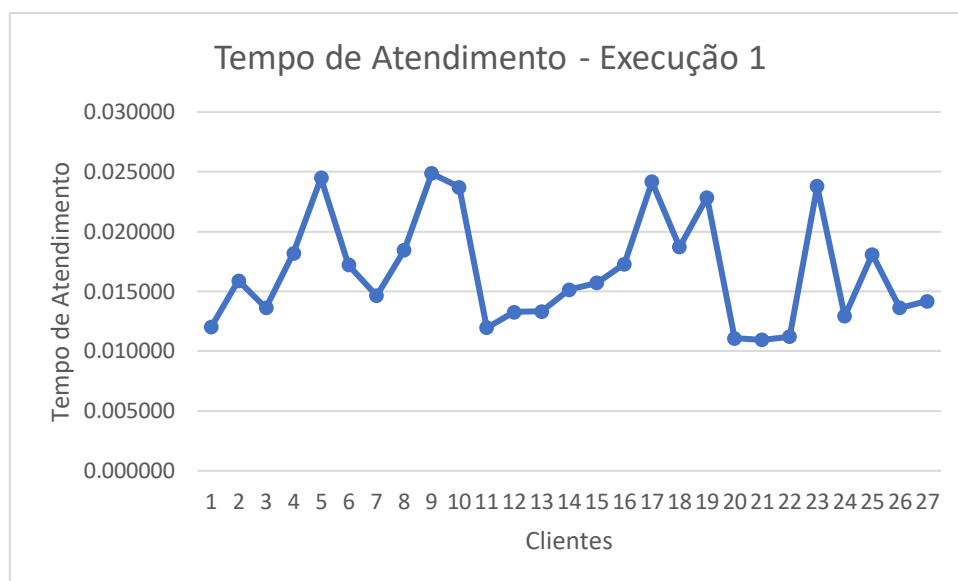


Tabela 6- Tempo de atendimento por cliente na primeira execução

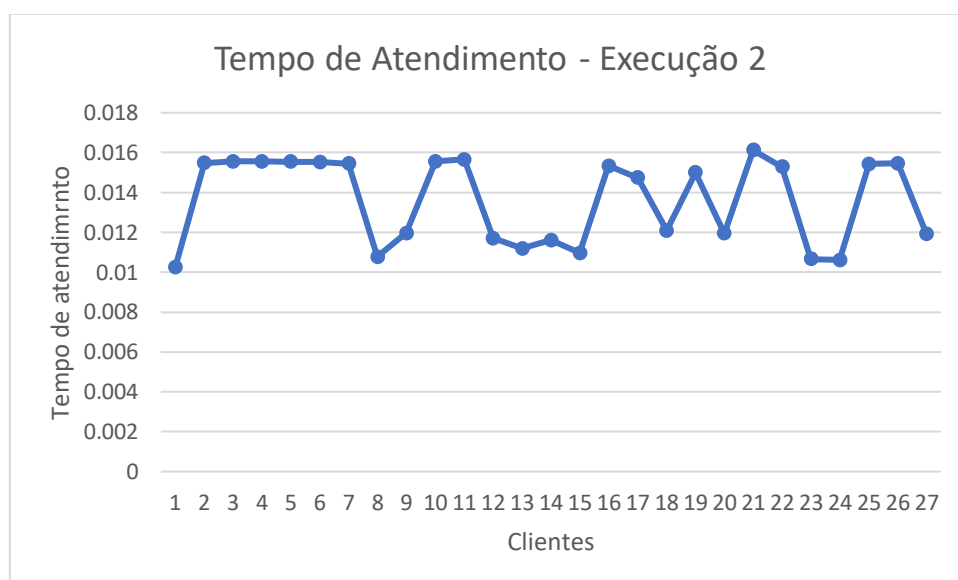


Tabela 7- Tempo de atendimento por cliente na segunda execução



Tabela 8- Tempo de atendimento por cliente na terceira execução

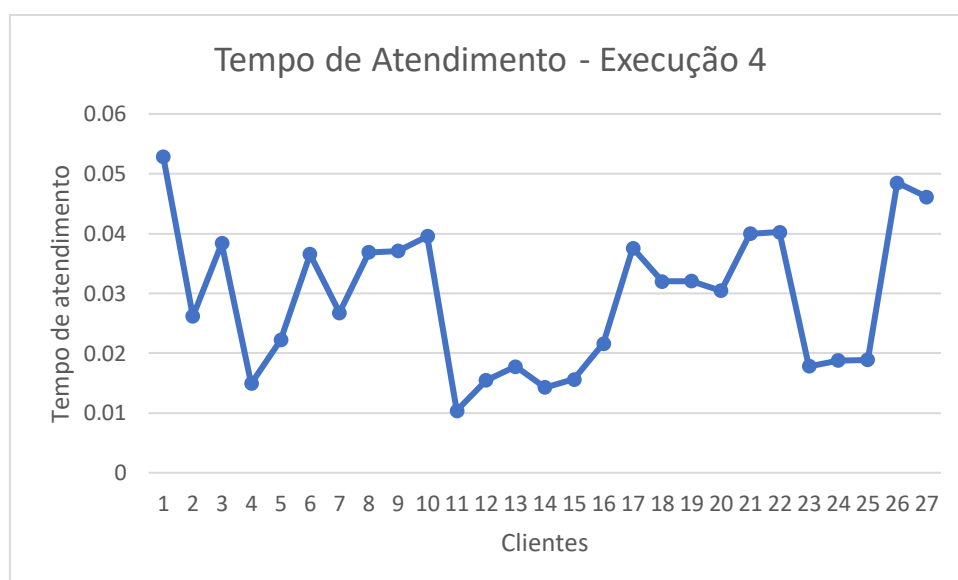


Tabela 9- Tempo de atendimento por cliente na quarta execução

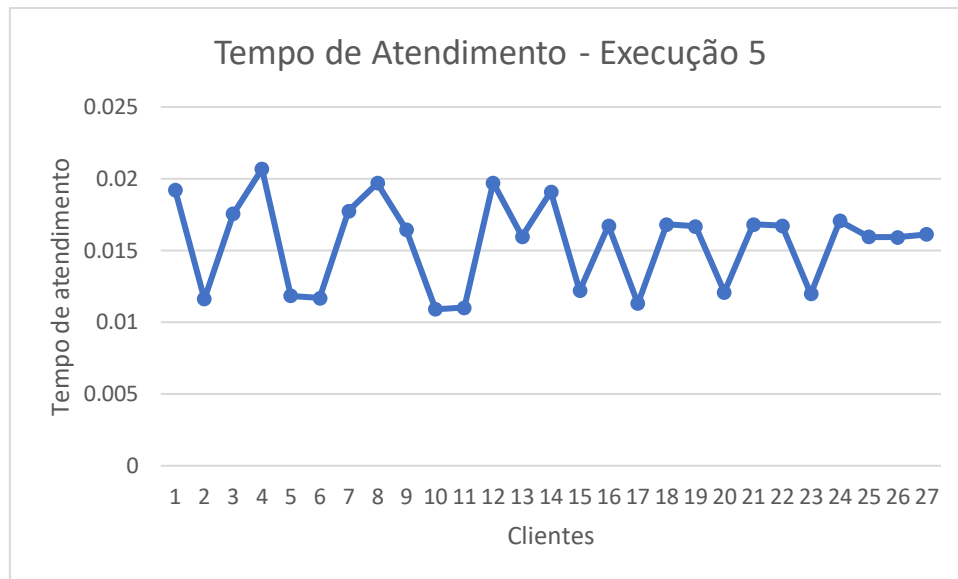


Tabela 10- Tempo de atendimento por cliente na quinta execução

Assim como no primeiro programa, as primeiras threads, não foram as primeiras a serem executadas. Mais uma vez, assim como visto nos outros experimentos ao longo do semestre, verificamos que tanto o tamanho da string quanto o ordem de inicialização das threads, não influencia o tempo de execução de tal processo. O SO, tem suas políticas próprias de escalonamento e o usuário não tem controle sobre elas. Portanto, um programa que está rodando em background, pode ter maior prioridade se comparado com as threads do programa. Com isso, não sabemos quando os processos serão executados e nem a ordem de execução dos mesmos.

Portanto, os resultados estão de acordo com as políticas do SO. As threads possuem diferentes tempos de execução, que não estão relacionados com a ordem das mesmas.

Conclusão

Como citado anteriormente, o experimento possibilitou o uso de todos os mecanismos vistos no semestre. Foi importante para que os conceitos fossem melhor compreendidos pelos alunos. Mais uma vez, foi possível observar, que o escalonamento do SO, e possíveis programas consumindo a CPU alteram os resultados esperados do programa.

Outro ponto importante foi a possibilidade de revisão de algumas funções vistas ao longo do semestre. No primeiro programa, foi possível a revisão de funções relacionadas a fila de mensagens e memória compartilhada como: `shmget()`, `shmat()`, `msgget()`, entre outras. No segundo programa, todas as funções relacionadas a threads, mutex e semáforos também puderam ser revisadas, contribuindo para a conclusão do experimento.