

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Институт №8 “Компьютерные науки и прикладная математика”
Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №2 по курсу
«Операционные системы»

Группа: М8О-212БВ-24

Студент: Мышакин М.С.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 18.10.25

Москва, 2025

Постановка задачи

Вариант 5.

Отсортировать массив целых чисел при помощи четно-нечетной сортировки Бетчера.

Общий метод и алгоритм решения

Для сортировки массива целых чисел используется четно-нечетная сортировка Бетчера (Odd-Even Transposition Sort). Алгоритм основан на последовательных сравнениях и обменах соседних элементов в массиве.

Принцип работы алгоритма:

1. Массив проходит через четные и нечетные фазы:
 - Четная фаза: сравниваются пары элементов с индексами (0,1), (2,3), ...
 - Нечетная фаза: сравниваются пары элементов с индексами (1,2), (3,4), ...
2. Если пара элементов расположена в неправильном порядке, значения меняются местами.
3. Процесс повторяется до тех пор, пока массив не станет полностью отсортированным.

Многопоточность:

- Каждое сравнение пары элементов выполняется в отдельном потоке.
- Введено ограничение на максимальное количество одновременно работающих потоков, которое задаётся при запуске программы.
- С помощью глобального счетчика `active_threads` и мьютекса `mutex` программа контролирует, чтобы количество потоков не превышало заданное.

Описание работы программы

Запуск программы:

```
./program <размер массива> <максимальное число потоков>
```

Пример:

```
./program 20 4
```

- Размер массива = 20
- Максимальное число потоков = 4

Структура программы:

1. Генерация массива случайных целых чисел.
2. Функция `odd_even_sort`:
 - Организация четных и нечетных фаз.
 - Создание потоков `pthread` для каждой пары элементов.
 - Контроль количества активных потоков через `active_threads` и `mutex`.
3. Ждем завершения каждого потока в фазе через `pthread_join`.
4. Вывод исходного и отсортированного массива.
5. Измерение времени сортировки с помощью `clock()`.

Контроль количества потоков:

- Перед запуском каждого потока программа проверяет глобальный счетчик активных потоков.
- Если достигнут лимит, поток ждет, пока другой завершится.

Пример вывода программы:

Исходный массив:

523 12 845 234 7 98 ...

Отсортированный массив:

7 12 34 98 123 234 ...

Время сортировки: 0.034 секунд

Код программы

main.c

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#include <time.h>

#define MAX_SIZE 1000

typedef struct {
    int *arr;
    int i;
    int j;
} ThreadData;

int active_threads = 0;
int max_threads = 2;
pthread_mutex_t mutex;

void *compare_and_swap(void *arg) {
    ThreadData *data = (ThreadData *)arg;

    pthread_mutex_lock(&mutex);
    while (active_threads >= max_threads) {
        pthread_mutex_unlock(&mutex);
        usleep(100);
        pthread_mutex_lock(&mutex);
    }
    active_threads++;
    pthread_mutex_unlock(&mutex);

    if (data->arr[data->i] > data->arr[data->j]) {
        int temp = data->arr[data->i];
        data->arr[data->i] = data->arr[data->j];
        data->arr[data->j] = temp;
    }
}
```

```

    data->arr[data->i] = data->arr[data->j];
    data->arr[data->j] = temp;
}

pthread_mutex_lock(&mutex);
active_threads--;
pthread_mutex_unlock(&mutex);

free(data);
return NULL;
}

void odd_even_sort(int *arr, int n) {
    int sorted = 0;

    while (!sorted) {
        sorted = 1;
        pthread_t threads[MAX_SIZE];
        int thread_count = 0;

        for (int i = 0; i <= n - 2; i += 2) {
            ThreadData *data = malloc(sizeof(ThreadData));
            data->arr = arr;
            data->i = i;
            data->j = i + 1;
            pthread_create(&threads[thread_count++], NULL, compare_and_swap,
data);
        }

        for (int i = 0; i < thread_count; i++)
            pthread_join(threads[i], NULL);
    }
}

```

```

thread_count = 0;

for (int i = 1; i <= n - 2; i += 2) {

    ThreadData *data = malloc(sizeof(ThreadData));

    data->arr = arr;
    data->i = i;
    data->j = i + 1;

    pthread_create(&threads[thread_count++], NULL, compare_and_swap,
data);

}

for (int i = 0; i < thread_count; i++) {
    pthread_join(threads[i], NULL);
}

for (int i = 0; i < n - 1; i++) {
    if (arr[i] > arr[i + 1]) {
        sorted = 0;
        break;
    }
}

}

int main(int argc, char *argv[]) {
    if (argc < 3) {
        printf("Использование: %s <размер массива> <макс потоков>\n", argv[0]);
        return 1;
    }

    int n = atoi(argv[1]);
    max_threads = atoi(argv[2]);
}

```

```
if (n > MAX_SIZE) n = MAX_SIZE;

int arr[MAX_SIZE];
srand(time(NULL));
for (int i = 0; i < n; i++)
    arr[i] = rand() % 1000;

printf("Исходный массив:\n");
for (int i = 0; i < n; i++)
    printf("%d ", arr[i]);
printf("\n");

pthread_mutex_init(&mutex, NULL);

clock_t start = clock();
odd_even_sort(arr, n);
clock_t end = clock();

pthread_mutex_destroy(&mutex);

printf("Отсортированный массив:\n");
for (int i = 0; i < n; i++)
    printf("%d ", arr[i]);
printf("\n");

printf("Время сортировки: %.3f секунд\n", (double)(end - start) /
CLOCKS_PER_SEC);

return 0;
}
```

Протокол работы программы

```
misha@misha-VirtualBoxUbuntu:~/Documents/cApps/OS-LABS/laba2$ ./main 10 1
```

Исходный массив:

770 298 624 208 178 752 725 876 780 701

Отсортированный массив:

178 208 298 624 701 725 752 770 780 876

Время сортировки: 0.009 секунд

```
misha@misha-VirtualBoxUbuntu:~/Documents/cApps/OS-LABS/laba2$ ./main 10 10
```

Исходный массив:

482 640 628 516 951 257 447 735 458 111

Отсортированный массив:

111 257 447 458 482 516 628 640 735 951

Время сортировки: 0.012 секунд

```
misha@misha-VirtualBoxUbuntu:~/Documents/cApps/OS-LABS/laba2$ ./main 50 1
```

Исходный массив:

212 428 15 346 468 907 514 200 740 755 392 875 819 854 542 728 60 84 249 562 736 843

487 825 193 96 799 387 143 562 327 355 991 694 53 811 601 919 12 694 675 756 569 494

963 464 223 23 548 824

Отсортированный массив:

12 15 23 53 60 84 96 143 193 200 212 223 249 327 346 355 387 392 428 464 468 487 494
514 542 548 562 569 601 675 694 694 728 736 740 755 756 799 811 819 824 825 843 854
875 907 919 963 991

Время сортировки: 0.145 секунд

```
misha@misha-VirtualBoxUbuntu:~/Documents/cApps/OS-LABS/laba2$ ./main 50 10
```

Исходный массив:

888 446 124 680 110 467 777 349 927 769 116 7 218 153 827 978 750 268 235 735 543 946

660 560 915 492 270 712 606 1 965 846 799 441 878 909 908 7 258 188 776 374 195 995 879

374 973 981 995 560

Отсортированный массив:

1 7 7 110 116 124 153 188 195 218 235 258 268 270 349 374 374 441 446 467 492 543 560
560 606 660 680 712 735 750 769 776 777 799 827 846 878 879 888 908 909 915 927 946 965
973 978 981 995 995

Время сортировки: 0.155 секунд

```
misha@misha-VirtualBoxUbuntu:~/Documents/cApps/OS-LABS/laba2$ ./main 500 1
```

Исходный массив:

806 505 13 182 913 169 261 725 13 449 712 929 750 163 512 262 911 280 535 125 807 557

882 816 980 848 347 41 997 874 522 803 731 887 985 644 56 599 721 69 48 433 351 150

949 863 412 860 496 299 986 655 857 220 471 837 420 818 230 417 44 752 573 775 639 910

771 48 509 493 469 909 278 820 59 579 36 823 440 532 123 778 539 980 350 10 169 771

829 399 188 225 503 113 353 494 24 124 542 885 617 364 795 248 184 854 827 572 30 619

456 153 749 995 133 100 6 302 871 187 701 411 412 556 877 765 50 901 242 945 138 859

309 933 459 845 140 287 418 170 258 874 323 8 870 456 108 228 110 331 767 811 94 179

367 323 297 769 576 539 714 715 750 375 648 562 221 140 849 991 310 459 217 985 467

439 793 927 667 903 610 434 66 57 966 785 380 615 555 957 506 621 672 608 997 672 170

218 813 371 561 475 831 778 813 650 218 606 578 885 510 540 672 576 597 990 362 330

605 269 287 463 890 311 71 887 335 242 457 148 965 18 976 796 797 789 447 367 395 377

252 257 917 276 186 867 266 900 197 223 169 484 38 59 147 110 299 482 704 108 983 669
479 959 818 628 100 617 995 495 994 599 105 911 876 291 778 494 191 327 70 360 163 108
771 310 570 422 145 274 531 128 296 10 439 114 638 539 731 985 386 77 584 491 340 812
782 119 307 325 798 729 37 962 837 161 272 408 583 769 34 466 897 330 476 688 796 466
227 879 803 614 956 388 457 297 200 592 768 859 917 566 588 955 528 426 116 153 186 51
922 572 518 172 255 346 860 51 165 440 283 968 54 239 356 863 888 909 455 656 768 373
575 357 680 455 135 148 960 673 199 883 245 69 407 852 416 619 904 581 59 187 901 465
778 258 329 667 167 784 675 287 509 250 644 541 706 131 689 666 156 241 901 402 310
660 254 78 280 510 11 691 49 913 157 828 523 486 847 42 622 522 329 132 773 326 673
831 809 715 497 966 308 751 720 970 411 326 49 43 189 412 735 238 325 892 418 200 730
617 242 704 140 924 836 265 250 862 96 411 929 945 377 237 696 97 207 460 424 608 503
965 21 238 203 698 482 974 899 212 591 141 917 731 65 105 348 667 319 444

Отсортированный массив:

6 8 10 10 11 13 13 18 21 24 30 34 36 37 38 41 42 43 44 48 48 49 49 50 51 51 54 56 57 59
59 59 65 66 69 69 70 71 77 78 94 96 97 100 100 105 105 108 108 108 110 110 113 114 116
119 123 124 125 128 131 132 133 135 138 140 140 140 141 145 147 148 148 150 153 153 156
157 161 163 163 165 167 169 169 169 170 170 172 179 182 184 186 186 187 187 188 189 191
197 199 200 200 203 207 212 217 218 218 220 221 223 225 227 228 230 237 238 238 239 241
242 242 242 245 248 250 250 252 254 255 257 258 258 261 262 265 266 269 272 274 276 278
280 280 283 287 287 291 296 297 297 299 299 302 307 308 309 310 310 310 311 319 323
323 325 325 326 326 327 329 329 330 330 331 335 340 346 347 348 350 351 353 356 357 360
362 364 367 367 371 373 375 377 377 380 386 388 395 399 402 407 408 411 411 411 412 412
412 416 417 418 418 420 422 424 426 433 434 439 439 440 440 444 444 447 449 455 455 456 456
457 457 459 459 460 463 465 466 466 467 469 471 475 476 479 482 482 484 486 491 493 494
494 495 496 497 503 503 505 506 509 509 510 510 512 518 522 522 523 528 531 532 535 539
539 539 540 541 542 555 556 557 561 562 566 570 572 572 573 575 576 576 578 579 581 583
584 588 591 592 597 599 599 605 606 608 608 610 614 615 617 617 617 619 619 621 622 628
638 639 644 644 648 650 655 656 660 666 667 667 667 669 672 672 672 673 673 675 680 688
689 691 696 698 701 704 704 706 712 714 715 715 720 721 725 729 730 731 731 731 735 749
750 750 751 752 765 767 768 768 769 769 771 771 771 773 775 778 778 778 778 782 784 785
789 793 795 796 796 797 798 803 803 806 807 809 811 812 813 813 816 818 818 820 823 827
828 829 831 831 836 837 837 845 847 848 849 852 854 857 859 859 860 860 862 863 863 867
870 871 874 874 876 877 879 882 883 885 885 887 887 888 890 892 897 899 900 901 901 901
903 904 909 909 910 911 911 913 913 917 917 917 922 924 927 929 929 933 945 945 949 955
956 957 959 960 962 965 965 966 966 968 970 974 976 980 980 983 985 985 985 986 990 991
994 995 995 997 997

Время сортировки: 14.847 секунд

misha@misha-VirtualBoxUbuntu:~/Documents/cApps/OS-LABS/laba2\$./main 500 10

Исходный массив:

418 914 327 321 108 490 16 368 679 311 427 696 806 274 231 471 880 379 140 587 767 93
469 665 495 80 901 961 606 888 595 25 155 274 698 615 765 714 983 796 377 411 844 536
37 428 7 269 807 499 208 926 945 29 592 440 109 845 401 67 85 348 92 240 975 142 855
740 208 839 888 586 602 84 122 991 864 481 612 671 332 820 950 277 849 894 717 958 739
470 377 176 819 822 769 794 964 624 886 525 815 126 111 769 210 585 760 75 418 724 98
750 544 400 380 393 294 97 703 33 568 433 562 739 255 331 885 219 307 771 744 475 249
207 244 459 144 357 886 562 81 337 665 626 737 45 371 32 142 427 417 62 860 979 153 115
310 38 686 618 161 783 93 410 342 689 222 487 46 460 401 480 797 66 458 887 463 181 919
958 608 688 372 820 668 526 287 330 564 326 300 78 461 745 488 803 435 62 642 833 875
396 665 672 814 123 911 278 305 182 236 265 871 608 438 539 134 725 869 51 403 170 481

864 915 321 668 702 384 662 888 259 58 553 283 225 677 547 503 982 729 91 599 600 699
37 139 186 763 361 589 166 883 70 383 150 391 51 205 127 65 93 738 476 646 22 701 323
569 204 657 650 295 609 251 346 646 742 884 761 103 473 280 338 543 663 489 287 66 694
414 131 139 153 959 785 527 660 461 448 216 470 98 863 79 349 210 726 444 94 839 547
920 119 238 463 782 79 102 848 125 517 332 264 22 291 49 901 304 862 349 520 333 799
384 764 501 946 842 945 40 682 844 960 801 82 776 936 161 230 136 286 99 468 902 121
112 304 374 416 166 723 288 851 875 24 616 728 970 458 673 363 140 869 675 294 952 451
230 465 34 366 104 133 187 6 255 299 662 629 67 829 705 355 680 932 380 648 660 702 459
685 65 951 554 741 245 858 544 827 324 578 194 780 712 381 138 319 32 801 300 451 630
357 158 662 289 538 663 301 241 122 986 658 73 541 399 671 399 296 498 75 874 692 207
938 425 346 257 809 147 910 260 129 267 419 143 909 309 158 210 902 280 197 561 706 90
312 377 841 608 227 917 835 272 124 125 697 822 735 859 969 645 471 450 912 890 946 173
552 104 384 806 737 933 367 443 23 32 172

Отсортированный массив:

6 7 16 22 22 23 24 25 29 32 32 32 33 34 37 37 38 40 45 46 49 51 51 58 62 62 65 65 66
67 67 70 73 75 75 78 79 79 80 81 82 84 85 90 91 92 93 93 93 94 97 98 98 99 102 103 104
104 108 109 111 112 115 119 121 122 122 123 124 125 125 125 126 126 127 129 131 133 134 136 138
139 139 140 140 142 142 143 144 147 150 153 153 155 158 158 161 161 166 166 170 172 173
176 181 182 186 187 194 197 204 205 207 207 208 208 210 210 210 216 219 222 225 227 230
230 231 236 238 240 241 244 245 249 251 255 255 257 259 260 264 265 267 269 272 274 274
277 278 280 280 283 286 287 287 288 289 291 294 294 295 296 299 300 300 301 304 304 305
307 309 310 311 312 319 321 321 323 324 326 327 330 331 332 332 333 337 338 342 346 346
348 349 349 355 357 357 361 363 366 367 368 371 372 374 377 377 377 379 380 380 381 383
384 384 384 391 393 396 399 399 400 401 401 403 410 411 414 416 417 418 418 419 425 427
427 428 433 435 438 440 443 444 448 450 451 451 458 458 459 459 460 461 461 463 463 465
468 469 470 470 471 471 473 475 476 480 481 481 487 488 489 490 495 498 499 501 503 517
520 525 526 527 536 538 539 541 543 544 544 547 547 552 553 554 561 562 562 564 568 569
578 585 586 587 589 592 595 599 600 602 606 608 608 608 609 612 615 616 618 624 626 629
630 642 645 646 646 648 650 657 658 660 660 662 662 662 663 663 665 665 665 668 668 671
671 672 673 675 677 679 680 682 685 686 688 689 692 694 696 697 698 699 701 702 702 703
705 706 712 714 717 723 724 725 726 728 729 735 737 737 738 739 739 740 741 742 744 745
750 760 761 763 764 765 767 769 769 771 776 780 782 783 785 794 796 797 799 801 801 803
806 806 807 809 814 815 819 820 820 822 822 827 829 833 835 839 839 841 842 844 844 845
848 849 851 855 858 859 860 862 863 864 864 869 869 871 874 875 875 880 883 884 885 886
886 887 888 888 888 890 894 901 901 902 902 909 910 911 912 914 915 917 919 920 926 932
933 936 938 945 945 946 946 950 951 952 958 958 959 960 961 964 969 970 975 979 982 983
986 991

Время сортировки: 16.226 секунд

misha@misha-VirtualBoxUbuntu:~/Documents/cApps/OS-LABS/laba2\$./main 500 50

Исходный массив:

979 24 837 333 25 24 877 297 530 785 4 102 865 402 976 316 792 484 861 617 649 799 405
615 968 866 365 94 385 626 507 364 2 344 49 28 720 926 677 602 63 33 704 280 787 32 597
579 516 458 548 517 257 953 132 577 172 850 671 557 828 530 921 830 226 970 210 299 249
887 901 312 920 958 945 707 342 542 639 859 352 187 376 961 141 861 538 313 63 561 222
891 91 143 73 317 465 284 616 714 523 870 379 444 828 324 503 170 218 142 381 570 330
110 531 823 323 421 488 386 982 710 629 73 205 702 742 670 338 359 737 862 581 116 658
761 792 161 283 362 304 665 932 986 127 815 809 450 588 297 188 922 359 817 347 564 871
89 234 210 800 971 72 381 439 730 142 583 891 426 945 547 443 229 533 570 396 694 20
336 991 560 258 702 377 605 266 248 695 501 458 495 824 882 229 616 964 371 199 856 149

497 755 592 78 641 162 475 335 534 811 679 94 422 381 823 27 648 424 722 501 234 570
677 117 799 293 81 522 845 289 672 342 397 616 772 38 779 599 373 665 763 404 112 185
786 287 212 786 711 287 639 946 857 316 415 8 610 496 530 807 138 554 501 535 171 273
925 302 873 298 319 636 55 431 821 841 719 385 979 782 672 618 728 881 934 143 889 896
992 772 703 482 678 204 17 201 830 942 503 703 592 175 691 647 958 864 840 29 249 819
812 274 789 540 155 724 36 397 972 380 169 28 862 199 232 231 401 414 173 256 117 117
431 160 117 390 24 309 771 626 481 583 900 270 476 55 346 864 804 671 244 325 699 106
525 931 337 926 346 862 534 815 331 966 976 800 708 352 462 479 978 943 415 230 565 891
638 264 755 442 935 999 768 634 457 293 565 146 571 263 360 105 431 43 423 759 196 131
463 658 963 442 601 378 24 518 269 662 782 376 105 717 727 225 351 536 518 269 682 441
884 394 546 315 789 970 426 985 453 890 643 768 684 596 146 708 115 767 371 897 495 828
615 222 53 318 758 923 587 792 364 472 538 910 139 328 232 918 313 38 808 957 806 492
553 953 552 20 72 275 918 568 455 885 790 508 203 901 783 791 693 147 615 232 410 106
912 994 24 225 32 184 534 839 28 88 144 581 460

Отсортированный массив:

2 4 8 17 20 20 24 24 24 24 24 25 27 28 28 28 29 32 32 33 36 38 38 43 49 53 55 55 63 63
72 72 73 73 78 81 88 89 91 94 94 102 105 105 106 106 110 112 115 116 116 117 117 117
127 131 132 138 139 141 142 142 143 143 144 146 146 147 149 155 160 161 161 162 169 170 171
172 173 175 184 185 187 188 196 199 199 201 203 204 205 210 210 212 218 222 222 225 225
226 229 229 230 231 232 232 232 234 234 244 248 249 249 256 257 258 263 264 266 269 269
270 273 274 275 280 283 284 287 287 289 293 293 297 297 298 299 302 304 309 312 313 313
315 316 316 317 318 319 323 324 325 328 330 331 333 335 336 337 338 342 342 344 346 346
347 351 352 352 359 359 360 362 364 364 365 371 371 373 376 376 376 377 378 379 380 381 381
381 385 385 386 390 394 396 397 397 401 402 404 405 410 414 415 415 421 422 423 424 426
426 431 431 431 439 441 442 442 443 444 450 453 455 457 458 458 460 462 463 465 472 475
476 479 481 482 484 488 492 495 495 496 497 501 501 501 503 503 507 508 516 517 518 518
522 523 525 530 530 530 531 533 534 534 534 535 536 538 538 540 542 546 547 548 552 553
554 557 560 561 564 565 565 568 570 570 570 571 577 579 581 581 583 583 587 588 592 592
596 597 599 601 602 605 610 615 615 615 616 616 616 616 617 618 626 626 626 629 634 636 638 639
639 641 643 647 648 649 658 658 662 665 665 670 671 671 672 672 677 677 678 679 682 684
691 693 694 695 699 702 702 703 704 707 708 708 710 711 714 717 717 719 720 722 724 727
728 730 737 742 755 755 758 759 761 763 767 768 768 771 772 772 779 782 782 783 785 786
786 787 789 789 790 791 792 792 799 799 800 800 804 806 807 808 809 811 812 815 815
817 819 821 823 823 824 828 828 828 830 830 837 839 840 841 845 850 856 857 859 861 861
862 862 862 864 864 865 866 870 871 873 877 881 882 884 885 887 889 890 891 891 891 896
897 900 901 901 910 912 918 918 920 921 922 923 925 926 926 931 932 934 935 942 943 945
945 946 953 953 957 958 958 961 963 964 966 968 970 970 971 972 976 976 978 979 979 982
985 986 991 992 994 999

Время сортировки: 16.609 секунд

Отчет по эффективности и ускорению программы

После проведения тестирования программы, можно заметить, что при увеличении количества элементов, не учитывая количество потоков, программа работает дольше по времени. Если изменять количество потоков, то можно заметить, что скорость работы программы не уменьшается, как ожидалось, а увеличивается, поскольку на поток выделена маленькая работа по сравнению чисел, а время на инициализацию потока, проверки на превышение количества

потоков и другая внутренняя работа с потоками происходит дольше, чем это сравнение, отчего, соответственно, в данной задаче многопоточность является избыточной.

Вывод

В ходе выполнения лабораторной работы была составлена программа на языке Си, обрабатывающая данные в многопоточном режиме. Я получил опыт в управлении потоками в ОС и обеспечении синхронизации между потоками.