

CSE-467 - Parallel and Distributed Computing

Assignment No 2

Objective: Write a program for parallel vector addition by using multithreading.

The complexity of vector addition is $O(N)$ and the execution time depends on the vector length. In this assignment you will learn how to pass and return values from a thread and how logically work can be divided into multiple threads to be performed in parallel and reduce total execution time. You need to device some logic for load balancing.

Submit a word file containing all code, output snapshots and the comparison table and graph.

Input:

1. Vector Length: **N** (to be used to create and initialize 3 vectors)
2. No of Threads: **T**

Processing: Determine the execution time of vector addition with various values of **T** and **N** given in the table.

T_0 = get clock cycle/system time

Perform $V1 = V2 + V3$ with load balancing among **T** threads.

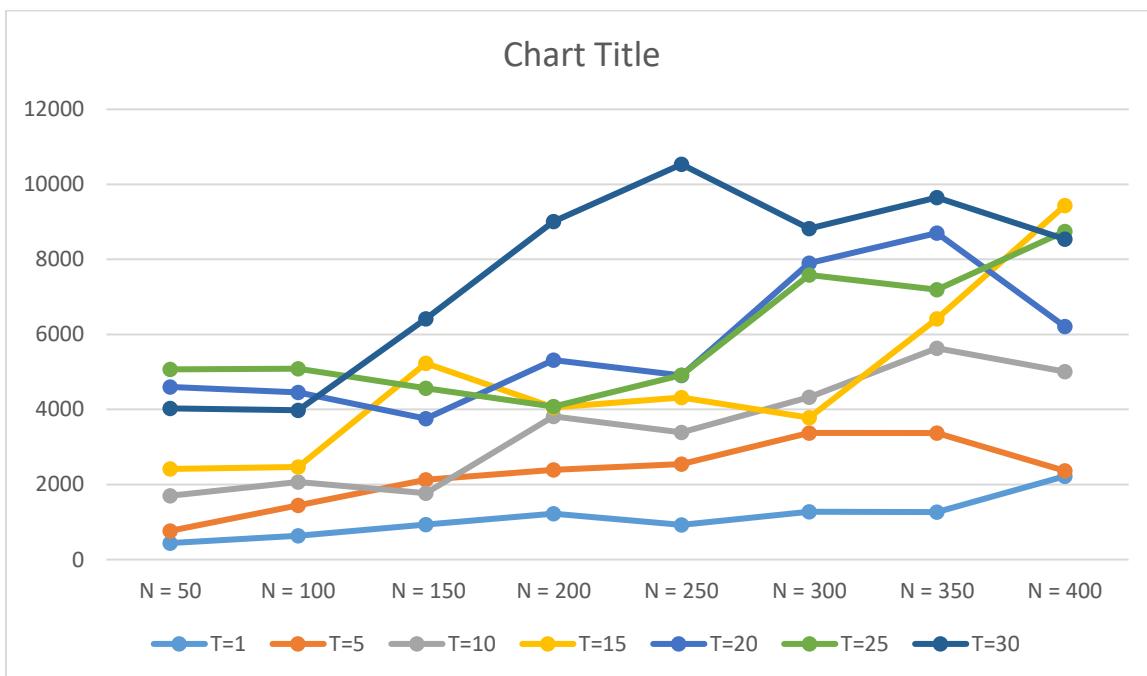
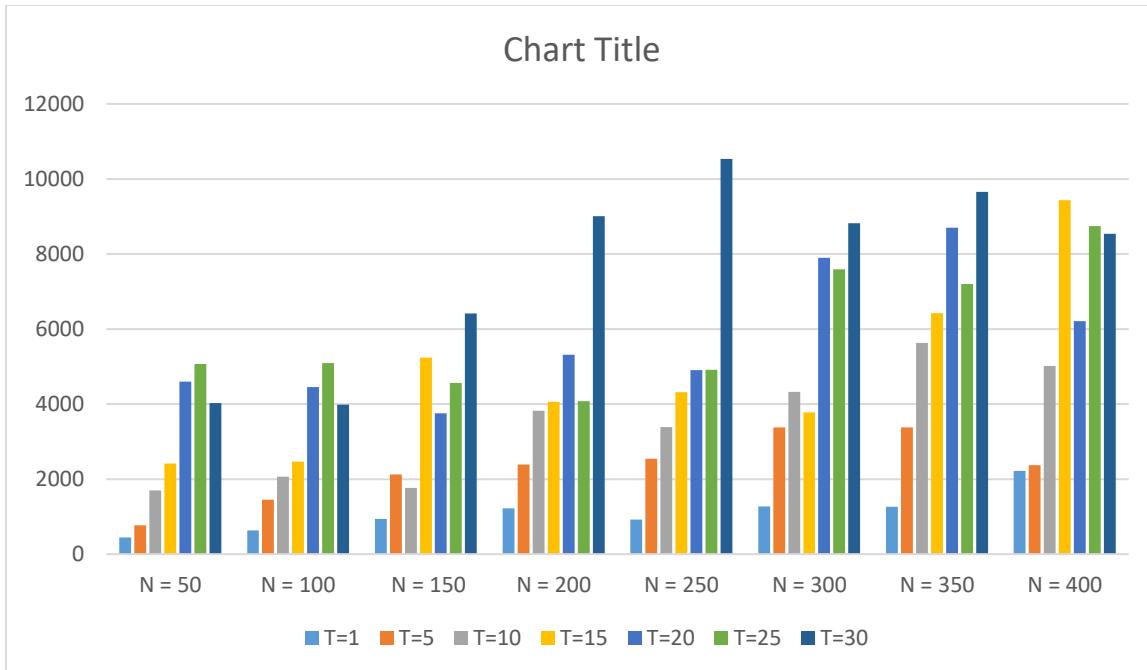
T_1 = get clock cycle/system time

Time elapsed = $T_1 - T_0$

Output: Time elapsed - Tables show the comparison

		N (vector size in thousands)							
		50	100	150	200	250	300	350	400
T (no of threads)	1	440	633	935	1220	923	1270	1267	2220
	5	766	1448	2128	2392	2544	3376	3374	3269
	10	1701	2065	1770	3819	3389	4327	5633	5014
	15	2415	2464	5235	4058	4317	3779	6421	9433
	20	4601	4453	3756	5317	4906	7901	8702	6209
	25	5068	5088	4566	4077	4916	7588	7196	8745
	30	4030	3980	6418	9009	10534	8823	9652	8540

Draw the Graph for above table.



Code:

```
#include <stddef.h>
#include <stdio.h>
#include <string.h>
#include <pthread.h>
#include <stdlib.h>
#include <time.h>

int N = 50;
int T = 1;
int* vector_1;
int* vector_2;
int* vector_res;

vector_Sum(int TID){
    int WL = (int) (N/T);
    int SID = TID * WL;

    for (int i = SID ; i < SID+WL ; i++){
        vector_res[i] = vector_1[i] + vector_2[i];
    }
}

int main()
{
    pthread_t *t1;
    vector_1 = (int*) malloc(N * sizeof(int));
    vector_2 = (int*) malloc(N * sizeof(int));
```

```

vector_res = (int*) malloc(N * sizeof(int));

for(int i = 0 ; i< N; i++){
    vector_1[i] = i;
    vector_2[i] = i;
}

double start = clock();

for(int i = 0 ; i < T ; i++){
    int tid = (int) malloc(sizeof(int));
    tid = i;
    pthread_create(&t1,NULL,vector_Sum,tid);
}

for(int i = 0 ; i < T ; i++){
    int tid = (int) malloc(sizeof(int));
    tid = i;
    pthread_join(t1,NULL);
}

if( N%T != 0){
    for (int i = ((int)(N/T))*T ; i < N ; i++){
        vector_res[i] = vector_1[i] + vector_2[i];
    }
}

double end = clock();
double x = (double) (end - start);
printf("Resultant Vector:\n");
for(int i = 0 ; i < N ; i++){
    printf("%d ",vector_res[i]);
}
printf("\nClock ticks = ");
printf("%lf \n" , x);

return 0;
}

```

Screen Shots:

T=1 , N = 50

The screenshot shows a Linux desktop environment with a terminal window titled "Assignment2.c". The code in the terminal is as follows:

```

1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 50000;
10 int T = 1;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17 int WL = (int) (N/T);
18 int SID = TID * WL;
19
20 for (int i = SID ; i < SID+WL ; i++){
21     vector_res[i] = vector_1[i] + vector_2[i];
22 }
23
24
25
26 int main(){
27
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int i = 0 ; i < T ; i++){
42         int tid = (int) malloc(sizeof(int));
43         tid = i;
44         pthread_create(&t1,NULL,vector_Sum,tid);
45     }
46
47     for(int i = 0 ; i < T ; i++){

```

The terminal output shows a series of random integers from 9953 to 9958, followed by "Clock ticks = 40.000000" and "oem@AliHaider:~\$".

T = 1 , N = 100

The screenshot shows a Linux desktop environment with a terminal window titled "Assignment2.c". The code is identical to the one above, but with N set to 100000. The terminal output is extremely long, consisting of a single line of approximately 100,000 random integers ranging from 0 to 99998, followed by "Clock ticks = 63.000000" and "oem@AliHaider:~\$".

T = 1 , N = 150

The terminal window shows the following C code:

```
#include <stddef.h>
#include <stdio.h>
#include <string.h>
#include <pthread.h>
#include <stdlib.h>
#include <time.h>

int N = 150000;
int T = 1;
int* vector_1;
int* vector_2;
int* vector_res;
int i;
int vector_Sum(int TID){
    int WL = (int)(N/T);
    int SID = TID * WL;
    for (int i = SID ; i < SID+WL ; i++){
        vector_res[i] = vector_1[i] + vector_2[i];
    }
}
int main()
{
    pthread_t *t1;
    vector_1 = (int*) malloc(N * sizeof(int));
    vector_2 = (int*) malloc(N * sizeof(int));
    vector_res = (int*) malloc(N * sizeof(int));
    for(int i = 0 ; i < N ; i++){
        vector_1[i] = i;
        vector_2[i] = i;
    }
    double start = clock();
    for(int i = 0 ; i < T ; i++){
        int tid = (int) malloc(sizeof(int));
        tid = i;
        pthread_create(&t1,NULL,vector_Sum,tid);
    }
    for(int i = 0 ; i < T ; i++){

```

The terminal output shows the sum of two vectors of size 150,000. The output is a long string of numbers starting with 66 299580 299519 299512 299514 299516 299518 299520 299522 299524 299526 299528 299530 299532 299534 299536 299538 299540 299542 299544 299546 299548 299550 299552 299554 299556 299558 299560 299562 299564 299566 299568 299570 299572 299574 299576 299578 299580 299582 299584 299586 299588 299590 299592 299594 299596 299598 299600 299602 299604 299606 299608 299610 299612 299614 299616 299618 29962 299624 299626 299628 299630 299632 299634 299636 299638 299640 299642 299644 299646 299648 299650 299652 299654 299656 299658 299660 299662 299664 299666 299668 299670 299672 299674 299676 299678 299680 299682 299684 299686 299688 299690 299692 299694 299696 299698 299700 299702 299704 299706 299708 299710 299712 299714 299716 299718 299720 299722 299724 299726 299728 299730 299732 299734 299736 299738 299740 299742 299744 299746 299748 299750 299752 299754 299756 299758 299760 299762 299764 299766 299768 299770 299772 299774 299776 299778 299780 299782 299784 299786 299788 299790 299792 299794 299796 299798 299800 299802 299804 299806 299808 299810 299812 299814 299816 299818 299820 299822 299824 299826 299828 299830 299832 299834 299836 299838 299840 299842 299844 299846 299848 299850 299852 299854 299856 299858 299860 299862 299864 299866 299868 299870 299872 299874 299876 299878 299880 299882 299884 299886 299888 299890 299892 299894 299896 299898 299900 299902 299904 299906 299908 299910 299912 299914 299916 299918 299920 299922 299924 299926 299928 299930 299932 299934 299936 299938 299940 299942 299944 299946 299948 299950 299952 299954 299956 299958 299960 299962 299964 299966 299968 299970 299972 299974 299976 299978 299980 299982 299984 299986 299988 299990 299992 299994 299996 299998 299999 299999 Clock ticks = 935.000000

Terminal prompt: oem@AliHaider: ~

T = 1 , N = 200

The terminal window shows the same C code as the previous screenshot, but for N=200. The output is a long string of numbers starting with 66 399508 399510 399512 399514 399516 399518 399520 399522 399524 399526 399528 399530 399532 399534 399536 399538 399540 399542 399544 399546 399548 399550 399552 399554 399556 399558 399560 399562 399564 399566 399568 399570 399572 399574 399576 399578 399580 399582 399584 399586 399588 399590 399592 399594 399596 399598 399600 399602 399604 399606 399608 399610 399612 399614 399616 399618 39962 399622 399624 399626 399628 399630 399632 399634 399636 399638 399640 399642 399644 399646 399648 399650 399652 399654 399656 399658 399660 399662 399664 399666 399668 399670 399672 399674 399676 399678 399680 399682 399684 399686 399688 399690 399692 399694 399696 399698 399700 399702 399704 399706 399708 399710 399712 399714 399716 399718 399720 399722 399724 399726 399728 399730 399732 399734 399736 399738 399740 399742 399744 399746 399748 399750 399752 399754 399756 399758 399760 399762 399764 399766 399768 399770 399772 399774 399776 399778 399780 399782 399784 399786 399788 399790 399792 399794 399796 399798 399800 399802 399804 399806 399808 399810 399812 399814 399816 399818 399820 399822 399824 399826 399828 399830 399832 399834 399836 399838 399840 399842 399844 399846 399848 399850 399852 399854 399856 399858 399860 399862 399864 399866 399868 399870 399872 399874 399876 399878 399880 399882 399884 399886 399888 399890 399892 399894 399896 399898 399900 399902 399904 399906 399908 399910 399912 399914 399916 399918 399920 399922 399924 399926 399928 399930 399932 399934 399936 399938 399940 399942 399944 399946 399948 399950 399952 399954 399956 399958 399960 399962 399964 399966 399968 399970 399972 399974 399976 399978 399980 399982 399984 399986 399988 399990 399992 399994 399996 399998 399999 399999 Clock ticks = 1226.000000

Terminal prompt: oem@AliHaider: ~

T = 1 , N = 250

The terminal window displays a large sequence of numbers, likely the output of a computation. The command prompt is 'oeng@AllHalder: ~'. The desktop environment includes icons for various applications like a browser, file manager, and system tools.

```
06 399580 399510 399512 399514 399516 399518 399520 399522 399524 399526 399528  
399530 399532 399534 399536 399538 399540 399542 399544 399546 399548 399550 399552  
399554 399556 399558 399560 399562 399564 399566 399568 399570 399572 399574  
399576 399578 399580 399582 399584 399586 399588 399590 399592 399594 399596 399598  
399599 399600 399602 399604 399606 399608 399610 399612 399614 399616 399618 399620  
399622 399624 399626 399628 399630 399632 399634 399636 399638 399640 399642 399644  
399646 399648 399650 399652 399654 399656 399658 399660 399662 399664 399666 399668  
399670 399672 399674 399676 399678 399680 399682 399684 399686 399688  
399690 399692 399694 399696 399698 399700 399702 399704 399706 399708 399710 399712  
399714 399716 399718 399720 399722 399724 399726 399728 399730 399732 399734  
399736 399738 399740 399742 399744 399746 399748 399750 399752 399754 399756 399758  
399760 399762 399764 399766 399768 399770 399772 399774 399776 399778 399780  
0 399782 399784 399786 399788 399790 399792 399794 399796 399798 399799 399800 399802  
99804 399806 399808 399810 399812 399814 399816 399818 399820 399822 399824 399826  
26 399828 399830 399832 399834 399836 399838 399840 399842 399844 399846 399848  
399850 399852 399854 399856 399858 399860 399862 399864 399866 399868 399870 399872  
399874 399876 399878 399880 399882 399884 399886 399888 399890 399892 399894 399896  
399898 399899 399900 399902 399904 399906 399908 399910 399912 399914 399916 399918  
399920 399922 399924 399926 399928 399930 399932 399934 399936 399938 399940 399942  
0 399944 399946 399948 399950 399952 399954 399956 399958 399960 399962 399964 399966  
399968 399970 399972 399974 399976 399978 399980 399982 399984 399986 399988 399990  
86 399988 399990 399992 399994 399996 399998 399999 399999 399999 399999 399999 399999  
Clock ticks = 923.000000  
oeng@AllHalder: $
```

T = 1 , N = 300

The terminal window displays a large sequence of numbers, likely the output of a computation. The command prompt is 'oeng@AllHalder: ~'. The desktop environment includes icons for various applications like a browser, file manager, and system tools.

```
06 599580 599510 599512 599514 599516 599518 599520 599522 599524 599526 599528  
599530 599532 599534 599536 599538 599540 599542 599544 599546 599548 599550 599552  
599554 599556 599558 599560 599562 599564 599566 599568 599570 599572 599574  
599576 599578 599580 599582 599584 599586 599588 599590 599592 599594 599596 599598  
599599 599600 599602 599604 599606 599608 599610 599612 599614 599616 599618 599620  
0 599622 599624 599626 599628 599630 599632 599634 599636 599638 599640 599642 599644  
599646 599648 599650 599652 599654 599656 599658 599660 599662 599664 599666 599668  
599669 599670 599672 599674 599676 599678 599680 599682 599684 599686 599688 599690  
712 599714 599716 599718 599720 599722 599724 599726 599728 599730 599732 599734  
599736 599738 599740 599742 599744 599746 599748 599750 599752 599754 599756 599758  
9758 599760 599762 599764 599766 599768 599770 599772 599774 599776 599778 599780  
0 599782 599784 599786 599788 599790 599792 599794 599796 599798 599799 599800 599802 599804  
999806 599808 599810 599812 599814 599816 599818 599820 599822 599824 599826 599828  
26 599828 599830 599832 599834 599836 599838 599840 599842 599844 599846 599848  
599850 599852 599854 599856 599858 599860 599862 599864 599866 599868 599870 599872  
872 599874 599876 599878 599880 599882 599884 599886 599888 599890 599892 599894 599896  
599898 599899 599900 599902 599904 599906 599908 599910 599912 599914 599916 599918  
599920 599922 599924 599926 599928 599930 599932 599934 599936 599938 599940 599942  
0 599944 599946 599948 599950 599952 599954 599956 599958 599960 599962 599964 599966  
399964 399966 399968 399970 399972 399974 399976 399978 399980 399982 399984 399986  
86 399988 399990 399992 399994 399996 399998 399999 399999 399999 399999 399999 399999  
Clock ticks = 1273.000000  
oeng@AllHalder: $
```

T = 1 , N = 350

The terminal window shows a series of floating-point numbers representing the results of the vector summation. The code in the editor is identical to the one shown for T=1, N=300.

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 350000;
10 int T = 1;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int i = 0 ; i < T ; i++){
42         int tid = (int) malloc(sizeof(int));
43         tid = i;
44         pthread_create(&t1,NULL,vector_Sum,tid);
45     }
46
47     for(int i = 0 ; i < T ; i++){
```

Clock ticks = 1267.000000
oem@AliHaider: ~

T = 1 , N = 400

The terminal window shows a series of floating-point numbers representing the results of the vector summation. The code in the editor is identical to the one shown for T=1, N=300.

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 400000;
10 int T = 1;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int i = 0 ; i < T ; i++){
42         int tid = (int) malloc(sizeof(int));
43         tid = i;
44         pthread_create(&t1,NULL,vector_Sum,tid);
45     }
46
47     for(int i = 0 ; i < T ; i++){
```

Clock ticks = 2226.000000
oem@AliHaider: ~

T = 5 , N = 50

The terminal window displays a long sequence of numerical values ranging from 24 to 99426, likely representing the results of a computation. The code editor window shows a C++ program named 'Assignment2c' that performs a vector sum operation using threads.

```
24 99426 99428 99430 99432 99434 99436 99438 99440 99442 99444 99446 99448 99450  
99452 99454 99456 99458 99460 99462 99464 99466 99468 99470 99472 99474 99476 9  
9478 99480 99482 99484 99486 99488 99490 99492 99494 99496 99498 99500 99502 995  
64 99506 99508 99510 99512 99514 99516 99518 99520 99522 99524 99526 99528 99530  
99532 99534 99536 99538 99540 99542 99544 99546 99548 99550 99552 99554 99556 9  
9558 99558 99562 99564 99566 99568 99570 99572 99574 99576 99578 99580 99582 995  
84 99584 99586 99588 99590 99592 99594 99596 99598 99600 99602 99604 99606 99608 99610  
99612 99614 99616 99618 99620 99622 99624 99626 99628 99630 99632 99634 99636 9  
9638 99640 99642 99644 99646 99648 99650 99652 99654 99656 99658 99660 99662 996  
64 99666 99668 99670 99672 99674 99676 99678 99680 99682 99684 99686 99688 9968  
99692 99694 99696 99698 99700 99702 99704 99706 99708 99710 99712 99714 99716 9  
9718 99720 99722 99724 99726 99728 99730 99732 99734 99736 99738 99740 99742 997  
44 99744 99746 99748 99750 99752 99754 99756 99758 99760 99762 99764 99766 9976  
99772 99774 99776 99778 99780 99782 99784 99786 99788 99790 99792 99794 99796 9  
9798 99800 99802 99804 99806 99808 99810 99812 99814 99816 99818 99820 99822 998  
24 99826 99828 99830 99832 99834 99836 99838 99840 99842 99844 99846 99848 99850  
99852 99854 99856 99858 99860 99862 99864 99866 99868 99870 99872 99874 99876 9  
9878 99880 99882 99884 99886 99888 99890 99892 99894 99896 99898 99900 99902 9990  
61 99904 99906 99908 99910 99912 99914 99916 99918 99920 99922 99924 99926 99928 999  
99931 99934 99936 99938 99940 99942 99944 99946 99948 99950 99952 99954 99956 9  
9958 99960 99962 99964 99966 99968 99970 99972 99974 99976 99978 99980 99982 9998  
84 99986 99988 99990 99992 99994 99996 99998 99999 99999 99999 99999 99999 99999  
Clock ticks = 666.000000  
oem@AllHaider: ~
```

T = 5 , N = 100

The terminal window displays a long sequence of numerical values ranging from 06 to 99508, likely representing the results of a computation. The code editor window shows a C++ program named 'Assignment2c' that performs a vector sum operation using threads.

```
06 199508 199510 199512 199514 199516 199518 199520 199522 199524 199526 199528  
199530 199532 199534 199536 199538 199540 199542 199544 199546 199548 199550 19  
552 199554 199556 199558 199560 199562 199564 199566 199568 199570 199572 199574 199576 19  
199578 199580 199582 199584 199586 199588 199590 199592 199594 199596 19  
9598 199600 199602 199604 199606 199608 199610 199612 199614 199616 199618 19962  
0 199622 199624 199626 199628 199630 199632 199634 199636 199638 199640 199642 1  
99644 199646 199648 199650 199652 199654 199656 199658 199660 199662 199664 1996  
66 199668 199670 199672 199674 199676 199678 199680 199682 199684 199686 199688  
199690 199692 199694 199696 199698 199700 199702 199704 199706 199708 199710 199712  
199714 199716 199718 199720 199722 199724 199726 199728 199730 199732 199734 199736  
199738 199730 199740 199742 199744 199746 199748 199750 199752 199754 199756 19  
9758 199766 199762 199764 199766 199768 199770 199772 199774 199776 199778 19978  
0 199782 199784 199786 199788 199790 199792 199794 199796 199798 199800 199802 1  
99804 199806 199808 199810 199812 199814 199816 199818 199820 199822 199824 1998  
26 199826 199828 199830 199832 199834 199836 199838 199840 199842 199844 199846 19984  
48 199850 199852 199854 199856 199858 199860 199862 199864 199866 199868 199870 19  
872 199874 199876 199878 199880 199882 199884 199886 199888 199890 199892 199894  
199896 199898 199900 199902 199904 199906 199908 199910 199912 199914 199916 19  
9918 199920 199922 199924 199926 199928 199930 199932 199934 199936 199938 19994  
0 199942 199944 199946 199948 199950 199952 199954 199956 199958 199960 199962 1  
99964 199966 199968 199970 199972 199974 199976 199978 199980 199982 199984 1999  
86 199988 199990 199992 199994 199996 199998 199999 199999 199999 199999 199999 19999  
Clock ticks = 1448.000000  
oem@AllHaider: ~
```

T = 5 , N = 150

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 150000;
10
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_sum(int TID){
17     int WL = (int)(N/T);
18     int SID = TID * WL;
19
20     for (int l = SID ; l < SID+WL ; l++){
21         vector_res[l] = vector_1[l] + vector_2[l];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int l = 0 ; l < T ; l++){
42         int tid = (int) malloc(sizeof(int));
43         tid = l;
44         pthread_create(&t1,NULL,vector_sum,tid);
45     }
46
47     for(int l = 0 ; l < T ; l++)
```

oem@AliHaider: ~

```
06 299508 299510 299512 299514 299516 299518 299520 299522 299524 299526 299528
299530 299520 299524 299524 299525 299525 299526 299526 299527 299528 299529
552 299554 299556 299558 299560 299562 299562 299564 299564 299566 299568 299570 299572 299574
299576 299578 299580 299582 299584 299586 299588 299588 299590 299592 299594 299596
9598 299600 299602 299604 299606 299608 299610 299612 299614 299616 299618 29962
0 299622 299624 299626 299628 299630 299632 299634 299636 299638 299640 299642 299644
9944 299646 299648 299650 299652 299654 299656 299658 299660 299662 299664 299666 299668
66 299670 299672 299674 299676 299678 299680 299682 299684 299686 299688 299690 299692
299694 299696 299698 299700 299702 299704 299706 299708 299710 299712 299714 299716
712 299714 299716 299718 299720 299722 299724 299726 299728 299730 299732 299734
299736 299738 299740 299742 299744 299746 299748 299750 299752 299754 299756 29
9758 299760 299762 299764 299766 299768 299770 299772 299774 299776 299778 29978
0 299782 299784 299786 299788 299790 299792 299794 299796 299798 299800 299802 2
99804 299806 299808 299810 299812 299814 299816 299818 299820 299822 299824 299826 299828
26 299828 299830 299832 299834 299836 299838 299840 299842 299844 299846 299848
299850 299852 299854 299856 299858 299860 299862 299864 299866 299868 299870 29
872 299874 299876 299878 299880 299882 299884 299886 299888 299890 299892 299894 299896
299898 299900 299902 299904 299906 299908 299910 299912 299914 299916 299918 29
9918 299920 299922 299924 299926 299928 299930 299932 299934 299936 299938 29994
0 299940 299942 299944 299946 299948 299950 299952 299954 299956 299958 299960 299962 2
99964 299966 299968 299970 299972 299974 299976 299978 299980 299982 299984 299986
86 299988 299990 299992 299994 299996 299998 299999 299999 299999 299999 299999 299999
Clock ticks = 2128.000000
oem@AliHaider: $
```

T = 5 , N = 200

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 200000;
10
11 int T = 5;
12 int* vector_1;
13 int* vector_2;
14 int* vector_res;
15
16
17 vector_sum(int TID){
18     int WL = (int)(N/T);
19     int SID = TID * WL;
20
21     for (int l = SID ; l < SID+WL ; l++){
22         vector_res[l] = vector_1[l] + vector_2[l];
23     }
24 }
25
26
27 int main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i < N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int l = 0 ; l < T ; l++){
43         int tid = (int) malloc(sizeof(int));
44         tid = l;
45         pthread_create(&t1,NULL,vector_sum,tid);
46     }
47
48     for(int l = 0 ; l < T ; l++)
```

oem@AliHaider: ~

```
06 399508 399510 399512 399514 399516 399518 399520 399522 399524 399526 399528
399530 399532 399534 399536 399538 399540 399542 399544 399546 399548 399550 39
552 399554 399556 399558 399560 399562 399562 399564 399564 399566 399568 399570 399572 399574
399576 399578 399580 399582 399584 399586 399588 399588 399590 399592 399594 399596 39
9998 399600 399602 399604 399606 399608 399610 399612 399614 399616 399618 39962
0 399622 399624 399626 399628 399630 399632 399634 399636 399638 399640 399642 3
99644 399646 399648 399650 399652 399654 399656 399658 399660 399662 399664 39966
66 399668 399670 399672 399674 399676 399678 399680 399682 399684 399686 399688 399690
399692 399694 399696 399698 399700 399702 399704 399706 399708 399710 399712 399714
712 399714 399716 399718 399720 399722 399724 399726 399728 399730 399732 399734
399736 399738 399740 399742 399744 399746 399748 399750 399752 399754 399756 39
9758 399760 399762 399764 399766 399768 399770 399772 399774 399776 399778 39978
0 399782 399784 399786 399788 399790 399792 399794 399796 399798 399800 399802 3
99804 399806 399808 399810 399812 399814 399816 399818 399820 399822 399824 39982
26 399828 399830 399832 399834 399836 399838 399840 399842 399844 399846 399848
299850 399852 399854 399856 399858 399860 399862 399864 399866 399868 399870 399872
399874 399876 399878 399880 399882 399884 399886 399888 399890 399892 399894 399896
9918 399898 399900 399902 399904 399906 399908 399910 399912 399914 399916 399918 39992
0 399942 399944 399946 399948 399950 399952 399954 399956 399958 399960 399962 3
99964 399966 399968 399970 399972 399974 399976 399978 399980 399982 399984 399986
86 399988 399990 399992 399994 399996 399998 399999 399999 399999 399999 399999 399999
Clock ticks = 2392.000000
oem@AliHaider: $
```

T = 5 , N = 250

```
Assignment2.c
06 499508 499510 499512 499514 499516 499518 499520 499522 499524 499526 499528
499530 499532 499534 499536 499538 499540 499542 499544 499546 499548 499550 499552
552 499554 499556 499558 499560 499562 499564 499566 499568 499570 499572 499574
499576 499578 499580 499582 499584 499586 499588 499590 499592 499594 499596 499598
9598 499600 499602 499604 499606 499608 499610 499612 499614 499616 499618 49962
0 499620 499622 499624 499626 499628 499630 499632 499634 499636 499638 499640 499642
99644 499646 499648 499650 499652 499654 499656 499658 499660 499662 499664 49966
66 499668 499670 499672 499674 499676 499678 499680 499682 499684 499686 499688
499690 499692 499694 499696 499698 499700 499702 499704 499706 499708 499710 499
712 499714 499716 499718 499720 499722 499724 499726 499728 499730 499732 499734
499736 499738 499740 499742 499744 499746 499748 499750 499752 499754 499756 499
78 499758 499760 499762 499764 499766 499768 499770 499772 499774 499776 499778 49978
0 499782 499784 499786 499788 499790 499792 499794 499796 499798 499800 499802 4
99884 499886 499888 499890 499891 499892 499894 499896 499898 499899 499900 499902 499904
26 499828 499830 499832 499834 499836 499838 499840 499842 499844 499846 499848 499850
499852 499854 499856 499858 499860 499862 499864 499866 499868 499870 499872 499874
872 499874 499876 499878 499880 499882 499884 499886 499888 499890 499892 499894
499896 499898 499899 499900 499902 499904 499906 499908 499910 499912 499914 499916
9918 499920 499922 499924 499926 499928 499930 499932 499934 499936 499938 49994
0 499942 499944 499946 499948 499950 499952 499954 499956 499958 499960 499962 499964
499966 499968 499970 499972 499974 499976 499978 499980 499982 499984 499986 499988
86 499988 499989 499990 499991 499992 499993 499994 499995 499996 499997 499998 499999
Clock ticks = 2544.000000
oempAllHolder: -5
```

T = 5 , N = 300

```
Assignment2.c
06 599508 599510 599512 599514 599516 599518 599520 599522 599524 599526 599528
599530 599532 599534 599536 599538 599540 599542 599544 599546 599548 599550 599552
552 599554 599556 599558 599560 599562 599564 599566 599568 599570 599572 599574
599576 599578 599580 599582 599584 599586 599588 599590 599592 599594 599596 599598
9598 599600 599602 599604 599606 599608 599610 599612 599614 599616 599618 59962
0 599620 599622 599624 599626 599628 599630 599632 599634 599636 599638 599640 599642
99644 599646 599648 599650 599652 599654 599656 599658 599660 599662 599664 59966
66 599668 599670 599672 599674 599676 599678 599680 599682 599684 599686 599688
599690 599692 599694 599696 599698 599700 599702 599704 599706 599708 599710 599
712 599714 599716 599718 599720 599722 599724 599726 599728 599730 599732 599734
599736 599738 599740 599742 599744 599746 599748 599750 599752 599754 599756 599
78 599758 599760 599762 599764 599766 599768 599770 599772 599774 599776 599778 59978
0 599782 599784 599786 599788 599790 599792 599794 599796 599798 599800 599802 5
99884 599886 599888 599890 599891 599892 599894 599896 599898 599899 599900 599902
26 599828 599830 599832 599834 599836 599838 599840 599842 599844 599846 599848 599850
599852 599854 599856 599858 599860 599862 599864 599866 599868 599870 599872 599874
872 599874 599876 599878 599880 599882 599884 599886 599888 599890 599892 599894
599896 599898 599899 599900 599902 599904 599906 599908 599910 599912 599914 599916
9918 599920 599922 599924 599926 599928 599930 599932 599934 599936 599938 59994
0 599942 599944 599946 599948 599950 599952 599954 599956 599958 599960 599962 599964
599966 599968 599970 599972 599974 599976 599978 599980 599982 599984 599986 599988
86 599988 599989 599990 599991 599992 599993 599994 599995 599996 599997 599998 599999
Clock ticks = 3379.000000
oempAllHolder: -5
```

T = 5 , N = 350

The terminal window displays a sequence of approximately 350 integers, each consisting of 10 digits. The code editor window shows a C++ program named `Assignment2.c` that performs a vector sum operation using threads. The program includes headers for `<csignal.h>`, `<stdio.h>`, `<string.h>`, `<pthread.h>`, `<stdlib.h>`, and `<time.h>`. It defines constants `N` and `WL`, initializes vectors `vector_1` and `vector_2`, and calculates the sum in `vector_res` using a parallel loop with `pthread_t`.

```
1 #include <csignal.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8 int N = 350000;
9 int T = 5;
10 int* vector_1;
11 int* vector_2;
12 int* vector_res;
13
14
15 vector_sum(int TID){
16     int WL = (int)(N/T);
17     int SID = TID * WL;
18
19     for (int i = SID ; i < SID+WL ; i++){
20         vector_res[i] = vector_1[i] + vector_2[i];
21     }
22 }
23
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int i = 0 ; i < T ; i++){
42         int tid = (int) malloc(sizeof(int));
43         tid = i;
44         pthread_create(&t1,NULL,vector_sum,tid);
45     }
46
47     for(int i = 0 ; i < T ; i++){
48
49 }
```

```
oe@AllHaider:~$ 06 699508 699510 699512 699514 699516 699518 699520 699522 699524 699526 699528
699530 699532 699534 699536 699538 699540 699542 699544 699546 699548 699550 699
552 699554 699556 699558 699560 699562 699564 699566 699568 699570 699572 699574
699576 699578 699580 699582 699584 699586 699588 699590 699592 699594 699596 699
598 699606 699608 699609 699610 699612 699614 699616 699618 69962
0 699620 699622 699624 699626 699628 699630 699632 699634 699636 699638 699640 699642 699644 699646 699648 699650 699652 699654 699656 699658 699660 699662 699664 6996
66 699668 699670 699672 699674 699676 699678 699680 699682 699684 699686 699688
699690 699692 699694 699696 699698 699700 699702 699704 699706 699708 699710 699712 699714 699716 699718 699720 699722 699724 699726 699728 699730 699732 699734
699736 699738 699740 699742 699744 699746 699748 699750 699752 699754 699756 699758 699760 699762 699764 699766 699768 699770 699772 699774 699776 699778 699780 699782 699784 699786 699788 699790 699792 699794 699796 699798 699800 699802 699804 699806 699808 699810 699812 699814 699816 699818 699820 699822 699824 699826 699828 699830 699832 699834 699836 699838 699840 699842 699844 699846 699848 699850 699852 699854 699856 699858 699860 699862 699864 699866 699868 699870 699872 699874 699876 699878 699880 699882 699884 699886 699888 699890 699892 699894 699896 699898 699899 699900 699902 699904 699906 699908 699910 699912 699914 699916 699918 699920 699922 699924 699926 699928 699930 699932 699934 699936 699938 699940 699942 699944 699946 699948 699950 699952 699954 699956 699958 699960 699962 699964 699966 699968 699970 699972 699974 699976 699978 699980 699982 699984 699986 699988 699990 699992 699994 699996 699998 699999 699999
Clock ticks = 3374.000000
oe@AllHaider:~$
```

T = 5 , N = 400

The terminal window displays a sequence of approximately 400 integers, each consisting of 10 digits. The code editor window shows a C++ program named `Assignment2.c` that performs a vector sum operation using threads. The program includes headers for `<csignal.h>`, `<stdio.h>`, `<string.h>`, `<pthread.h>`, `<stdlib.h>`, and `<time.h>`. It defines constants `N` and `WL`, initializes vectors `vector_1` and `vector_2`, and calculates the sum in `vector_res` using a parallel loop with `pthread_t`.

```
1 #include <csignal.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8 int N = 400000;
9 int T = 5;
10 int* vector_1;
11 int* vector_2;
12 int* vector_res;
13
14
15 vector_sum(int TID){
16     int WL = (int)(N/T);
17     int SID = TID * WL;
18
19     for (int i = SID ; i < SID+WL ; i++){
20         vector_res[i] = vector_1[i] + vector_2[i];
21     }
22 }
23
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int i = 0 ; i < T ; i++){
42         int tid = (int) malloc(sizeof(int));
43         tid = i;
44         pthread_create(&t1,NULL,vector_sum,tid);
45     }
46
47     for(int i = 0 ; i < T ; i++){
48
49 }
```

```
oe@AllHaider:~$ 06 799508 799510 799512 799514 799516 799518 799520 799522 799524 799526 799528
799530 799532 799534 799536 799538 799540 799542 799544 799546 799548 799550 799
552 799554 799556 799558 799560 799562 799564 799566 799568 799570 799572 799574
799576 799578 799580 799582 799584 799586 799588 799590 799592 799594 799596 799
598 799606 799608 799610 799612 799614 799616 799618 799620 799622 799624 799626 799628 799630 799632 799634 799636 799638 799640 799642 799644 799646 799648 799650 799652 799654 799656 799658 799660 799662 799664 799666 799668 799670 799672 799674 799676 799678 799680 799682 799684 799686 799688 799690 799692 799694 799696 799698 799699 799700 799702 799704 799706 799708 799710 799712 799714 799716 799718 799720 799722 799724 799726 799728 799730 799732 799734
799736 799738 799740 799742 799744 799746 799748 799750 799752 799754 799756 799758 799760 799762 799764 799766 799768 799770 799772 799774 799776 799778 799780 799782 799784 799786 799788 799790 799792 799794 799796 799798 799800 799802 799804 799806 799808 799810 799812 799814 799816 799818 799820 799822 799824 799826 799828 799830 799832 799834 799836 799838 799840 799842 799844 799846 799848 799850 799852 799854 799856 799858 799860 799862 799864 799866 799868 799870 799872 799874 799876 799878 799880 799882 799884 799886 799888 799890 799892 799894 799896 799898 799899 799900 799902 799904 799906 799908 799910 799912 799914 799916 799918 799920 799922 799924 799926 799928 799930 799932 799934 799936 799938 799940 799942 799944 799946 799948 799950 799952 799954 799956 799958 799960 799962 799964 799966 799968 799970 799972 799974 799976 799978 799980 799982 799984 799986 799988 799990 799992 799994 799996 799998 799999 799999
Clock ticks = 3629.000000
oe@AllHaider:~$
```

T = 10 , N = 50

The terminal window displays a long sequence of numerical values ranging from 94426 to 99586, likely representing the results of a parallel computation. The code editor window shows a C++ program named "Assignment2.c" that implements a parallel vector summation using multiple threads.

```

1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 50000;
10 int T = 10;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 int main()
28 {
29     pthread_t *tid;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&tid,NULL,vector_sum,tid);
46     }
47
48     for(int i = 0 ; i < T ; i++){
        
```

T = 10 , N = 100

The terminal window displays a massive sequence of numerical values, significantly longer than the previous output, representing the results of a parallel computation for N=100 and T=10. The code editor window shows the same C++ program as the first screenshot.

```

1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 100000;
10 int T = 10;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 int main()
28 {
29     pthread_t *tid;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&tid,NULL,vector_sum,tid);
46     }
47
48     for(int i = 0 ; i < T ; i++){
        
```

T = 10 , N = 150

The terminal window displays a sequence of approximately 150 integers, each consisting of 15 digits. The code editor window shows a C program named 'Assignment2.c' with the following content:

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8 int N = 150000;
9 int T = 10;
10 int* vector_1;
11 int* vector_2;
12 int* vector_res;
13
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int l = SID ; l < SID+WL ; l++){
21         vector_res[l] = vector_1[l] + vector_2[l];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int l = 0 ; l < N; l++){
35         vector_1[l] = l;
36         vector_2[l] = l;
37     }
38
39     double start = clock();
40
41     for(int l = 0 ; l < T ; l++){
42         int tid = (int) malloc(sizeof(int));
43         tid = l;
44         pthread_create(&t1,NULL,vector_Sum,tid);
45     }
46
47     for(int l = 0 ; l < T ; l++){
48 }
```

The desktop environment includes a taskbar with icons for various applications like a browser, file manager, and system tools. The system tray shows the date and time as 2/20/2023 at 12:03 AM.

T = 10 , N = 200

The terminal window displays a sequence of approximately 200 integers, each consisting of 15 digits. The code editor window shows a C program named 'Assignment2.c' with the following content:

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8 int N = 200000;
9 int T = 10;
10 int* vector_1;
11 int* vector_2;
12 int* vector_res;
13
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int l = SID ; l < SID+WL ; l++){
21         vector_res[l] = vector_1[l] + vector_2[l];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int l = 0 ; l < N; l++){
35         vector_1[l] = l;
36         vector_2[l] = l;
37     }
38
39     double start = clock();
40
41     for(int l = 0 ; l < T ; l++){
42         int tid = (int) malloc(sizeof(int));
43         tid = l;
44         pthread_create(&t1,NULL,vector_Sum,tid);
45     }
46
47     for(int l = 0 ; l < T ; l++){
48 }
```

The desktop environment includes a taskbar with icons for various applications like a browser, file manager, and system tools. The system tray shows the date and time as 2/20/2023 at 12:04 AM.

T = 10 , N = 250

```

Assignment2.c
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8 int N = 250000;
9 int T = 10;
10 int* vector_1;
11 int* vector_2;
12 int* vector_res;
13
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 int main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for (int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for (int l = 0 ; l < T ; l++){
43         int tid = (int) malloc(sizeof(int));
44         tid = l;
45         pthread_create(&t1,NULL,vector_Sum,tid);
46     }
47
48     for (int l = 0 ; l < T ; l++){
49
50         pthread_join(t1,&tid);
51         free(tid);
52     }
53
54     free(vector_1);
55     free(vector_2);
56     free(vector_res);
57
58     double end = clock();
59     double time = end - start;
60
61     printf("Time taken = %f\n", time);
62
63 }

```

Terminal Output:

```

oem@AliHalder:~$ ./Assignment2.c
Clock ticks = 3389.000000
oem@AliHalder:~$

```

T = 10 , N = 300

```

Assignment2.c
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 300000;
10 int T = 10;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 int main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for (int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for (int l = 0 ; l < T ; l++){
43         int tid = (int) malloc(sizeof(int));
44         tid = l;
45         pthread_create(&t1,NULL,vector_Sum,tid);
46     }
47
48     for (int l = 0 ; l < T ; l++){
49
50         pthread_join(t1,&tid);
51         free(tid);
52     }
53
54     free(vector_1);
55     free(vector_2);
56     free(vector_res);
57
58     double end = clock();
59     double time = end - start;
60
61     printf("Time taken = %f\n", time);
62
63 }

```

Terminal Output:

```

oem@AliHalder:~$ ./Assignment2.c
Clock ticks = 4327.000000
oem@AliHalder:~$

```

T = 10 , N = 350

The terminal window displays a long sequence of numerical values (ranging from 699508 to 99984) representing the output of a computation. The code editor window shows a C++ program named Assignment2.c that performs a vector sum operation using threads.

```
1 #include <csdef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 350000;
10 int T = 10;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_sum(int TID){
17     int WL = (int)(N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 int main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&t1,NULL,vector_sum,tid);
46     }
47
48     for(int i = 0 ; i < T ; i++){
49
50
51     }
52
53
54     pthread_join(t1, NULL);
55
56     free(vector_1);
57     free(vector_2);
58     free(vector_res);
59
60     free(tid);
61
62     double end = clock();
63
64     printf("Time taken : %f\n", (double)(end - start)/CLOCKS_PER_SEC);
65 }
```

T = 10 , N = 400

The terminal window displays a long sequence of numerical values (ranging from 699508 to 99984) representing the output of a computation. The code editor window shows a C++ program named Assignment2.c that performs a vector sum operation using threads.

```
1 #include <csdef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 400000;
10 int T = 10;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_sum(int TID){
17     int WL = (int)(N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 int main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&t1,NULL,vector_sum,tid);
46     }
47
48     for(int i = 0 ; i < T ; i++){
49
50
51     }
52
53
54     pthread_join(t1, NULL);
55
56     free(vector_1);
57     free(vector_2);
58     free(vector_res);
59
60     free(tid);
61
62     double end = clock();
63
64     printf("Time taken : %f\n", (double)(end - start)/CLOCKS_PER_SEC);
65 }
```

T = 15 , N = 50

The terminal window displays a sequence of approximately 1000 random integers ranging from 94426 to 99582. The code editor shows a C program named Assignment2.c that performs a vector sum operation using multiple threads.

```
Assignment2.c
```

```
1 #include <csddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 50000;
10 int T = 15;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *tis;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int i = 0 ; i < T ; i++){
42         int tid = (int) malloc(sizeof(int));
43         tid = i;
44         pthread_create(&tis[i],NULL,vector_Sum,tid);
45     }
46
47     for(int i = 0 ; i < T ; i++){
48
49     }
50 }
```

T = 15 , N = 100

The terminal window displays a sequence of approximately 1000 random integers ranging from 94426 to 99582. The code editor shows a C program named Assignment2.c that performs a vector sum operation using multiple threads.

```
Assignment2.c
```

```
1 #include <csddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 50000;
10 int T = 15;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *tis;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int i = 0 ; i < T ; i++){
42         int tid = (int) malloc(sizeof(int));
43         tid = i;
44         pthread_create(&tis[i],NULL,vector_Sum,tid);
45     }
46
47     for(int i = 0 ; i < T ; i++){
48
49     }
50 }
```

T = 15 , N = 150

The terminal window shows the following code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include <stdlib.h>
#include <time.h>

int N = 150000;
int T = 15;
int* vector_1;
int* vector_2;
int* vector_res;

vector_Sum(int TID){
    int WL = (int)(N/T);
    int SID = TID * WL;
    for (int i = SID ; i < SID+WL ; i++){
        vector_res[i] = vector_1[i] + vector_2[i];
    }
}

int main()
{
    pthread_t *t1;
    vector_1 = (int*) malloc(N * sizeof(int));
    vector_2 = (int*) malloc(N * sizeof(int));
    vector_res = (int*) malloc(N * sizeof(int));
    for(int i = 0 ; i < N; i++){
        vector_1[i] = i;
        vector_2[i] = i;
    }
    double start = clock();
    for(int i = 0 ; i < T ; i++){
        int tid = (int) malloc(sizeof(int));
        tid = i;
        pthread_create(&t1,NULL,vector_Sum,tid);
    }
    for(int i = 0 ; i < T ; i++){
        pthread_join(t1, NULL);
        free(tid);
    }
}
```

The terminal output shows the results of the computation:

```
oem@AliHaider: ~
06 2959508 299510 299512 299514 299516 299518 299520 299522 299524 299526 299528
299530 299532 299534 299536 299538 299540 299542 299544 299546 299548 299550 299
552 299554 299556 299558 299560 299562 299564 299566 299568 299570 299572 299574
399576 399578 399580 399582 399584 399586 399588 399590 399592 399594 399596 399
598 399600 399602 399604 399606 399608 399610 399612 399614 399616 399618 39962
0 299602 299624 299626 299628 299630 299632 299634 299636 299638 299640 299642 2
99644 299646 299648 299650 299652 299654 299656 299658 299660 299662 299664 2996
66 299668 299670 299672 299674 299676 299678 299680 299682 299684 299686 299688
299690 299692 299694 299696 299698 299700 299702 299704 299706 299708 299710 29
9712 299714 299716 299718 299720 299722 299724 299726 299728 299730 299732 299734 299736 299738 299740 299742 299744 299746 299748 299750 299752 299754 299756 29
9758 299760 299762 299764 299766 299768 299770 299772 299774 299776 299778 29978
0 299782 299784 299786 299788 299790 299792 299794 299796 299798 299800 299802 2
99804 299806 299808 299810 299812 299814 299816 299818 299820 299822 299824 2998
26 299828 299830 299832 299834 299836 299838 299840 299842 299844 299846 299848
299850 299852 299854 299856 299858 299860 299862 299864 299866 299868 299870 2998
87 299884 299886 299888 299890 299892 299894 299896 299898 299900 299902 299904 2999
299896 299898 299900 299902 299904 299906 299908 299910 299912 299914 299916 2999
918 299920 299922 299924 299926 299928 299930 299932 299934 299936 299938 29994
0 299942 299944 299946 299948 299950 299952 299954 299956 299958 299960 299962 2
99964 299966 299968 299970 299972 299974 299976 299978 299980 299982 299984 2999
86 299988 299990 299992 299994 299996 299998
Clock ticks = 5235.000000
oem@AliHaider: ~
```

T = 15 , N = 200

The terminal window shows the following code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include <stdlib.h>
#include <time.h>

int N = 200000;
int T = 15;
int* vector_1;
int* vector_2;
int* vector_res;

vector_Sum(int TID){
    int WL = (int)(N/T);
    int SID = TID * WL;
    for (int i = SID ; i < SID+WL ; i++){
        vector_res[i] = vector_1[i] + vector_2[i];
    }
}

int main()
{
    pthread_t *t1;
    vector_1 = (int*) malloc(N * sizeof(int));
    vector_2 = (int*) malloc(N * sizeof(int));
    vector_res = (int*) malloc(N * sizeof(int));
    for(int i = 0 ; i < N; i++){
        vector_1[i] = i;
        vector_2[i] = i;
    }
    double start = clock();
    for(int i = 0 ; i < T ; i++){
        int tid = (int) malloc(sizeof(int));
        tid = i;
        pthread_create(&t1,NULL,vector_Sum,tid);
    }
    for(int i = 0 ; i < T ; i++){
        pthread_join(t1, NULL);
        free(tid);
    }
}
```

The terminal output shows the results of the computation:

```
oem@AliHaider: ~
06 399508 399510 399512 399514 399516 399518 399520 399522 399524 399526 399528
399530 399532 399534 399536 399538 399540 399542 399544 399546 399548 399550 399
552 399554 399556 399558 399560 399562 399564 399566 399568 399570 399572 399574
399576 399578 399580 399582 399584 399586 399588 399590 399592 399594 399596 399
598 399600 399602 399604 399606 399608 399610 399612 399614 399616 399618 39962
0 399602 399624 399626 399628 399630 399632 399634 399636 399638 399640 399642 3
99644 399646 399648 399650 399652 399654 399656 399658 399660 399662 399664 3996
66 399668 399670 399672 399674 399676 399678 399680 399682 399684 399686 399688
399690 399692 399694 399696 399698 399700 399702 399704 399706 399708 399710 39
9712 399714 399716 399718 399720 399722 399724 399726 399728 399730 399732 399734 399736 399738 399740 399742 399744 399746 399748 399750 399752 399754 399756 39
9758 399760 399762 399764 399766 399768 399770 399772 399774 399776 399778 39978
0 399782 399784 399786 399788 399790 399792 399794 399796 399798 399800 399802 3
99804 399806 399808 399810 399812 399814 399816 399818 399820 399822 399824 3998
26 399828 399830 399832 399834 399836 399838 399840 399842 399844 399846 399848
399850 399852 399854 399856 399858 399860 399862 399864 399866 399868 399870 399
872 399874 399876 399878 399880 399882 399884 399886 399888 399890 399892 399894
399896 399898 399900 399902 399904 399906 399908 399910 399912 399914 399916 39
9918 399920 399922 399924 399926 399928 399930 399932 399934 399936 399938 3999
40 399942 399944 399946 399948 399950 399952 399954 399956 399958 399960 399962 3
99964 399966 399968 399970 399972 399974 399976 399978 399980 399982 399984 3999
86 399988 399990 399992 399994 399996 399998
Clock ticks = 4058.000000
oem@AliHaider: ~
```

T = 15, N = 250

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Assignment2.c" and the command entered is "oem@AllHaider:~". The terminal displays a very long command output consisting of a sequence of numbers (499508, 499532, etc.) separated by spaces. The desktop background is purple, and the taskbar at the bottom shows various application icons.

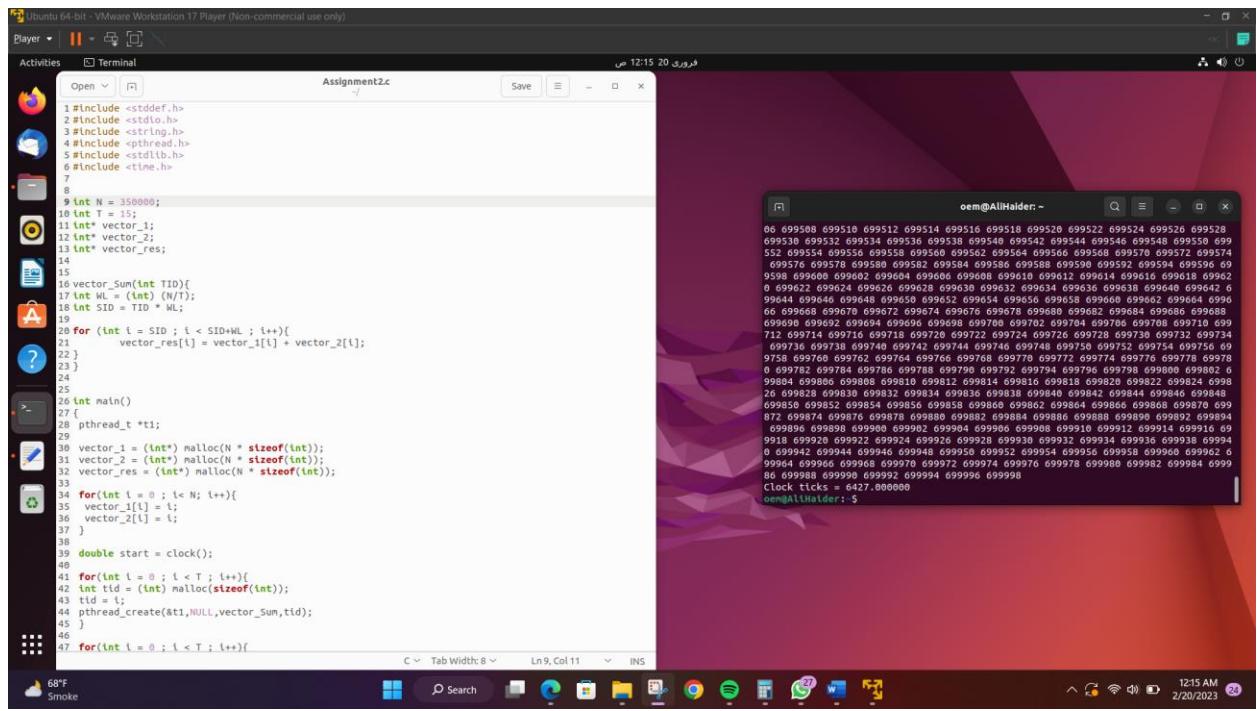
```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 350000;
10 int T = 15;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 int main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&t1,NULL,vector_Sum,tid);
46     }
47
48     for(int i = 0 ; i < T ; i++){
49
50
51
52
53
54
55
56
57
58
59
599508 499512 499512 499514 499516 499518 499520 499522 499524 499526 499528
599530 499532 499532 499534 499536 499538 499540 499542 499544 499546 499548 499550 499
552 499554 499554 499556 499558 499560 499562 499564 499566 499568 499570 499572 499574
499576 499578 499580 499582 499584 499586 499588 499590 499592 499594 499596 499598
9598 499600 499602 499604 499606 499608 499610 499612 499614 499616 499618 49962
0 499620 499622 499624 499626 499628 499630 499632 499634 499636 499638 499640 499642
99644 499646 499648 499650 499652 499654 499656 499658 499660 499662 499664 499666 499668
66 499668 499670 499672 499674 499676 499678 499680 499682 499684 499686 499688 499690
499692 499694 499696 499698 499699 499702 499704 499706 499708 499708 499710 499
712 499714 499716 499718 499720 499722 499724 499726 499728 499730 499732 499734
499736 499738 499740 499742 499744 499746 499748 499750 499752 499754 499756 499
758 499758 499760 499762 499764 499766 499768 499770 499772 499774 499776 499778 499780
0 499782 499784 499786 499788 499790 499792 499794 499796 499798 499800 499802 499804
499806 499808 499810 499812 499814 499816 499818 499820 499822 499824 499826
26 499828 499830 499832 499834 499836 499838 499840 499842 499844 499846 499848 499850
499852 499854 499856 499858 499860 499862 499864 499866 499868 499870 499872 499874
872 499874 499876 499878 499880 499882 499884 499886 499888 499890 499892 499894 499896
499898 499899 499901 499902 499904 499906 499908 499910 499912 499914 499916 499918
9918 499920 499922 499924 499926 499928 499930 499932 499934 499936 499938 499940
0 499942 499944 499946 499948 499950 499952 499954 499956 499958 499960 499962 499964
99964 499966 499968 499970 499972 499974 499976 499978 499980 499982 499984 499986
86 499988 499990 499992 499994 499996 499998 499999 499999 499999 499999 499999 499999
Clock ticks = 4317.000000
oem@AllHaider:~
```

T = 15 , N = 300

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Assignment2.c" and the command entered is "oem@AllHaider:~". The terminal displays a very long command output consisting of a sequence of numbers (599508, 599512, etc.) separated by spaces. The desktop background is purple, and the taskbar at the bottom shows various application icons.

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 300000;
10 int T = 15;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 int main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&t1,NULL,vector_Sum,tid);
46     }
47
48     for(int i = 0 ; i < T ; i++){
49
50
51
52
53
54
55
56
57
58
59
599508 599510 599512 599514 599516 599518 599520 599522 599524 599526 599528
599530 599532 599534 599536 599538 599540 599542 599544 599546 599548 599550 599
552 599554 599554 599556 599558 599560 599562 599564 599566 599568 599570 599572 599574
599576 599578 599580 599582 599584 599586 599588 599590 599592 599594 599596 599598
9598 599600 599602 599604 599606 599608 599610 599612 599614 599616 599618 59962
0 599620 599622 599624 599626 599628 599630 599632 599634 599636 599638 599640 599642
99644 599646 599648 599650 599652 599654 599656 599658 599660 599662 599664 599666 599668
66 599668 599670 599672 599674 599676 599678 599680 599682 599684 599686 599688 599690
599692 599694 599696 599698 599699 599702 599704 599706 599708 599710 599
712 599714 599716 599718 599720 599722 599724 599726 599728 599730 599732 599734
599736 599738 599740 599742 599744 599746 599748 599750 599752 599754 599756 599
758 599758 599760 599762 599764 599766 599768 599770 599772 599774 599776 599778 599780
0 599782 599784 599786 599788 599790 599792 599794 599796 599798 599800 599802 599804
99804 599806 599808 599810 599812 599814 599816 599818 599820 599822 599824 599826
26 599828 599830 599832 599834 599836 599838 599840 599842 599844 599846 599848 59984
599850 599852 599854 599856 599858 599860 599862 599864 599866 599868 599870 599872
872 599874 599876 599878 599880 599882 599884 599886 599888 599890 599892 599894
599896 599898 599900 599902 599904 599906 599908 599910 599912 599914 599916 599918
9918 599920 599922 599924 599926 599928 599930 599932 599934 599936 599938 599940
0 599942 599944 599946 599948 599950 599952 599954 599956 599958 599960 599962 599964
99964 599966 599968 599970 599972 599974 599976 599978 599980 599982 599984 599986
86 599988 599990 599992 599994 599996 599998 599999 599999 599999 599999 599999 599999
Clock ticks = 3779.000000
oem@AllHaider:~
```

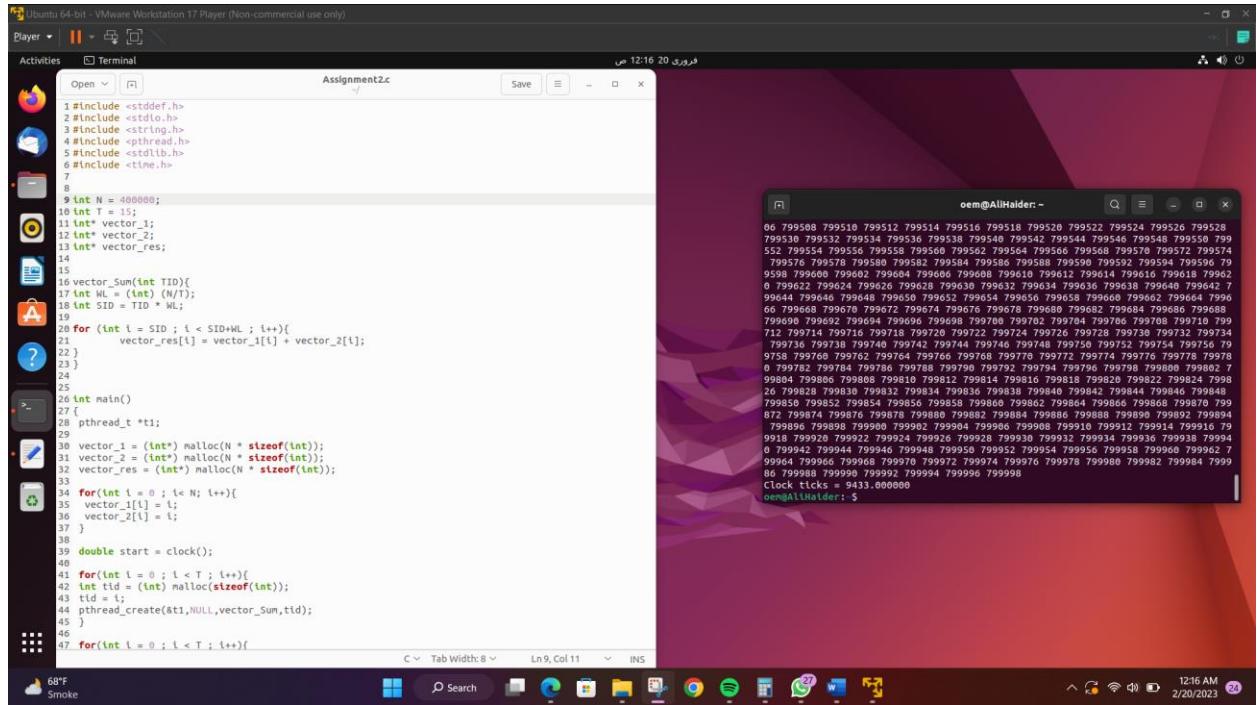
T = 15 , N = 350



The screenshot shows a Linux desktop environment with a terminal window and a code editor window. The terminal window displays a command-line session with numerous long numerical values. The code editor window shows a C++ program named 'Assignment2.c' with code related to vector operations and threads.

```
66 69958 699518 699512 699514 799516 799518 799520 799522 799524 799526 799528  
699530 699532 699534 699536 699538 699540 699542 699544 699546 699548 699550 699  
552 699554 699556 699558 699560 699562 699564 699566 699568 699570 699572 699574 699  
576 699578 699579 699580 699582 699584 699586 699588 699589 699590 699592 699594 699  
596 699596 699598 699599 699600 699601 699602 699603 699604 699605 699606 699607 699  
602 699608 699609 699610 699611 699612 699613 699614 699615 699616 699617 699618 699  
610 699619 699620 699621 699622 699623 699624 699625 699626 699627 699628 699629 699  
632 699630 699631 699632 699633 699634 699635 699636 699637 699638 699639 699640 699  
640 699641 699642 699643 699644 699645 699646 699647 699648 699649 699650 699651 699  
666 699652 699653 699654 699655 699656 699657 699658 699659 699660 699661 699662 699  
686 699663 699664 699665 699666 699667 699668 699669 699670 699671 699672 699673 699  
712 699714 699716 699718 699720 699722 699724 699726 699728 699730 699732 699734  
699736 699738 699740 699742 699744 699746 699748 699750 699752 699754 699756 699  
758 699760 699762 699764 699766 699768 699770 699772 699774 699776 699778 699780  
0 699782 699784 699786 699788 699790 699792 699794 699796 699798 699800 699802 6  
99884 699886 699888 699890 699891 699892 699893 699894 699895 699896 699897 699898  
26 699898 699899 699898 699897 699896 699895 699894 699893 699892 699891 699890 699  
699856 699857 699858 699859 699860 699861 699862 699863 699864 699865 699866 699867 699  
872 699868 699869 699870 699871 699872 699873 699874 699875 699876 699877 699878 699  
699879 699880 699881 699882 699883 699884 699885 699886 699887 699888 699889 699890  
9918 699920 699922 699924 699926 699928 699930 699932 699934 699936 699938 69994  
0 699942 699944 699946 699948 699950 699952 699954 699956 699958 699960 699962 6  
99964 699966 699968 699970 699972 699974 699976 699978 699980 699982 699984 699986  
86 699988 699989 699990 699992 699994 699996 699998 699999 699998 699992 699994 699996  
Clock ticks = 6427.000000  
oem@AllHolder:~
```

T = 15 , N = 400



The screenshot shows a Linux desktop environment with a terminal window and a code editor window. The terminal window displays a command-line session with numerous long numerical values. The code editor window shows a C++ program named 'Assignment2.c' with code related to vector operations and threads.

```
66 79958 799518 799512 799514 799516 799518 799520 799522 799524 799526 799528  
799530 799532 799534 799536 799538 799540 799542 799544 799546 799548 799550 799  
552 799554 799556 799558 799560 799562 799564 799566 799568 799570 799572 799574 799  
576 799578 799579 799580 799582 799584 799586 799588 799589 799590 799592 799594 799  
596 799596 799598 799599 799600 799601 799602 799603 799604 799605 799606 799607 799  
602 799608 799609 799610 799611 799612 799613 799614 799615 799616 799617 799618 799  
632 799619 799620 799621 799622 799623 799624 799625 799626 799627 799628 799629 799  
666 799630 799631 799632 799633 799634 799635 799636 799637 799638 799639 799640 799  
686 799641 799642 799643 799644 799645 799646 799647 799648 799649 799650 799651 799  
712 799714 799716 799718 799720 799722 799724 799726 799728 799730 799732 799734  
799736 799738 799740 799742 799744 799746 799748 799750 799752 799754 799756 799  
758 799760 799762 799764 799766 799768 799770 799772 799774 799776 799778 799780  
0 799782 799784 799786 799788 799790 799792 799794 799796 799798 799800 799802 7  
99884 799886 799888 799890 799891 799892 799893 799894 799895 799896 799897 799898  
26 799898 799899 799898 799897 799896 799895 799894 799893 799892 799891 799890 799  
799856 799857 799858 799859 799860 799861 799862 799863 799864 799865 799866 799867 799  
872 799874 799876 799878 799880 799882 799884 799886 799888 799889 799890 799892 799  
9918 799920 799922 799924 799926 799928 799930 799932 799934 799936 799938 79994  
0 799942 799944 799946 799948 799950 799952 799954 799956 799958 799960 799962 7  
99964 799966 799968 799970 799972 799974 799976 799978 799980 799982 799984 799986  
86 799988 799989 799990 799992 799994 799996 799998 799999 799998 799992 799994 799996  
Clock ticks = 9433.000000  
oem@AllHolder:~
```

T = 20 , N = 50

The terminal window displays a large list of numerical values ranging from 94426 to 99582, likely representing the results of a parallel computation. The code editor window shows a C++ program named Assignment2.c that implements a parallel vector sum operation using multiple threads.

```
#include <csdef.h>
#include <vector>
#include <iostream>
#include <string.h>
#include <pthread.h>
#include <stdlib.h>
#include <time.h>

int N = 500000;
int T = 20;
int* vector_1;
int* vector_2;
int* vector_res;
int SID;
int ML;
int vector_sum(int TID){
    int WL = (int) (N/T);
    int SID = TID * WL;
    for (int i = SID ; i < SID+WL ; i++){
        vector_res[i] = vector_1[i] + vector_2[i];
    }
}
int main()
{
    pthread_t *t1;
    double start = clock();
    vector_1 = (int*) malloc(N * sizeof(int));
    vector_2 = (int*) malloc(N * sizeof(int));
    vector_res = (int*) malloc(N * sizeof(int));
    for(int i = 0 ; i< N; i++){
        vector_1[i] = i;
        vector_2[i] = i;
    }
    for(int i = 0 ; i < T ; i++){
        int tid = (int) malloc(sizeof(int));
        tid = i;
        pthread_create(&t1,NULL,vector_sum,tid);
    }
    for(int i = 0 ; i < T ; i++){
        pthread_join(t1, NULL);
        free(tid);
    }
    double end = clock();
    cout << "Time taken : " << (end - start)/1000000.0 << endl;
    free(vector_1);
    free(vector_2);
    free(vector_res);
}
```

T = 20 , N = 100

The terminal window displays a massive list of numerical values, significantly longer than the previous output, indicating a larger dataset (N=100). The code editor window shows the same C++ program Assignment2.c as the first screenshot.

```
#include <csdef.h>
#include <vector>
#include <iostream>
#include <string.h>
#include <pthread.h>
#include <stdlib.h>
#include <time.h>

int N = 1000000;
int T = 20;
int* vector_1;
int* vector_2;
int* vector_res;
int SID;
int ML;
int vector_sum(int TID){
    int WL = (int) (N/T);
    int SID = TID * WL;
    for (int i = SID ; i < SID+WL ; i++){
        vector_res[i] = vector_1[i] + vector_2[i];
    }
}
int main()
{
    pthread_t *t1;
    double start = clock();
    vector_1 = (int*) malloc(N * sizeof(int));
    vector_2 = (int*) malloc(N * sizeof(int));
    vector_res = (int*) malloc(N * sizeof(int));
    for(int i = 0 ; i< N; i++){
        vector_1[i] = i;
        vector_2[i] = i;
    }
    for(int i = 0 ; i < T ; i++){
        int tid = (int) malloc(sizeof(int));
        tid = i;
        pthread_create(&t1,NULL,vector_sum,tid);
    }
    for(int i = 0 ; i < T ; i++){
        pthread_join(t1, NULL);
        free(tid);
    }
    double end = clock();
    cout << "Time taken : " << (end - start)/1000000.0 << endl;
    free(vector_1);
    free(vector_2);
    free(vector_res);
}
```

T = 20 , N = 150

The terminal window shows the following C code:

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 150000;
10 int T = 20;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int)(N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     double start = clock();
31
32     for(int i = 0 ; i < N; i++){
33         vector_1[i] = i;
34         vector_2[i] = i;
35     }
36
37     for(int i = 0 ; i < T ; i++){
38         int tid = (int) malloc(sizeof(int));
39         tid = i;
40         pthread_create(&t1,NULL,vector_Sum,tid);
41     }
42
43     for(int i = 0 ; i < T ; i++){
44         pthread_join(t1,&vector_res);
45     }
46
47 }
```

The terminal output shows the results of the computation:

```
oem@AliHaider: ~
06 299508 299510 299512 299514 299516 299518 299520 299522 299524 299526 299528
299530 299532 299534 299536 299538 299540 299542 299544 299546 299548 299549
552 299554 299556 299558 299560 299562 299564 299566 299568 299570 299572 299574
299576 299578 299580 299582 299584 299586 299588 299590 299592 299594 299596 29
0 299598 299600 299602 299604 299606 299608 299610 299612 299614 299616 299618 299620 299622 299624 299626 299628 299630 299632 299634 299636 299638 299640 299642 2
99644 299646 299648 299650 299652 299654 299656 299658 299660 299662 299664 2996
6 299666 299668 299670 299672 299674 299676 299678 299680 299682 299684 299686 299688
299690 299692 299694 299696 299698 299700 299702 299704 299706 299708 299710 299
712 299714 299716 299718 299720 299722 299724 299726 299728 299730 299732 299734
299736 299738 299740 299742 299744 299746 299748 299750 299752 299754 299756 29
9758 299760 299762 299764 299766 299768 299770 299772 299774 299776 299778 29978
0 299780 299782 299784 299786 299788 299790 299792 299794 299796 299798 29979
2 299804 299806 299808 299810 299812 299814 299816 299818 299820 299822 299824 2998
26 299828 299830 299832 299834 299836 299838 299840 299842 299844 299846 299848
299850 299852 299854 299856 299858 299860 299862 299864 299866 299868 299870 299
872 299874 299876 299878 299880 299882 299884 299886 299888 299889 299890 299892
9916 299894 299896 299898 299900 299902 299904 299906 299908 299910 299912 299914 299
9916 299922 299924 299926 299928 299930 299932 299934 299936 299938 29994
0 299940 299942 299944 299946 299948 299950 299952 299954 299956 299958 299960 299962 2
99964 299966 299968 299970 299972 299974 299976 299978 299980 299982 299984 29998
86 299986 299988 299990 299992 299994 299996 299998
Clock ticks = 1756.000000
oem@AliHaider: $
```

T = 20 , N = 200

The terminal window shows the following C code:

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 200000;
10 int T = 20;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int)(N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     double start = clock();
31
32     for(int i = 0 ; i < N; i++){
33         vector_1[i] = i;
34         vector_2[i] = i;
35     }
36
37     for(int i = 0 ; i < T ; i++){
38         int tid = (int) malloc(sizeof(int));
39         tid = i;
40         pthread_create(&t1,NULL,vector_Sum,tid);
41     }
42
43     for(int i = 0 ; i < T ; i++){
44         pthread_join(t1,&vector_res);
45     }
46
47 }
```

The terminal output shows the results of the computation:

```
oem@AliHaider: ~
06 399508 399510 399512 399514 399516 399518 399520 399522 399524 399526 399528
399530 399532 399534 399536 399538 399540 399542 399544 399546 399548 399550 39
552 399554 399556 399558 399560 399562 399564 399566 399568 399570 399572 399574
399576 399578 399580 399582 399584 399586 399588 399590 399592 399594 399596 39
0 399598 399600 399602 399604 399606 399608 399610 399612 399614 399616 399618 399620 399622 399624 399626 399628 399630 399632 399634 399636 399638 399640 399642 3
99644 399646 399648 399650 399652 399654 399656 399658 399660 399662 399664 3996
66 399668 399670 399672 399674 399676 399678 399680 399682 399684 399686 399688 39968
399690 399692 399694 399696 399698 399700 399702 399704 399706 399708 399710 399
712 399714 399716 399718 399720 399722 399724 399726 399728 399730 399732 399734
399736 399738 399740 399742 399744 399746 399748 399750 399752 399754 399756 39
9758 399760 399762 399764 399766 399768 399770 399772 399774 399776 399778 39978
0 399782 399784 399786 399788 399790 399792 399794 399796 399798 399800 399802 3
99804 399806 399808 399810 399812 399814 399816 399818 399820 399822 399824 3998
26 399828 399830 399832 399834 399836 399838 399840 399842 399844 399846 399848
399850 399852 399854 399856 399858 399860 399862 399864 399866 399868 399870 3998
399872 399874 399876 399878 399880 399882 399884 399886 399888 399890 399892 3998
94 399896 399898 399900 399902 399904 399906 399908 399910 399912 399914 399916 39
9918 399920 399922 399924 399926 399928 399930 399932 399934 399936 399938 39994
0 399942 399944 399946 399948 399950 399952 399954 399956 399958 399960 399962 3
99964 399966 399968 399970 399972 399974 399976 399978 399980 399982 399984 39998
86 399986 399988 399990 399992 399994 399996 399998
Clock ticks = 5319.000000
oem@AliHaider: $
```

T = 20 , N = 250

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 #define N = 250000;
10 int T = 20;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17 int WL = (int) (N/T);
18 int SID = TID * WL;
19
20 for (int i = SID ; i < SID+WL ; i++){
21     vector_res[i] = vector_1[i] + vector_2[i];
22 }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int l = 0 ; l < N; l++){
35         vector_1[l] = l;
36         vector_2[l] = l;
37     }
38
39     double start = clock();
40
41     for(int l = 0 ; l < T ; l++){
42         int tid = (int) malloc(sizeof(int));
43         tid = l;
44         pthread_create(&t1,NULL,vector_Sum,tid);
45     }
46
47     for(int l = 0 ; l < T ; l++){
48 }
```

```
6 499508 499510 499512 499514 499516 499518 499520 499522 499524 499526 499528
499530 499532 499534 499536 499538 499540 499542 499544 499546 499548 499550 499
52 499554 499556 499558 499560 499562 499564 499566 499568 499570 499572 499574
499576 499578 499580 499582 499584 499586 499588 499590 499592 499594 499596 49
58 499600 499602 499604 499606 499608 499610 499612 499614 499616 499618 49962
6 499622 499624 499626 499628 499630 499632 499634 499636 499638 499640 499642 49
64 499644 499646 499648 499650 499652 499654 499656 499658 499660 499662 499664 499
66 499666 499670 499672 499674 499676 499678 499680 499682 499684 499686 499688
499690 499692 499694 499696 499698 499700 499702 499704 499706 499708 499710 499
72 499714 499716 499718 499720 499722 499724 499726 499728 499730 499732 499734
499736 499738 499740 499742 499744 499746 499748 499750 499752 499754 499756 49
78 499768 499770 499772 499774 499776 499778 499780 499782 499784 499786 499788
0 499790 499792 499794 499796 499798 499799 499800 499802 499803 499804 499805 499806 49
9804 499806 499808 499810 499812 499814 499816 499818 499820 499822 499824 4998
26 499828 499830 499832 499834 499836 499838 499840 499842 499844 499846 499848
499856 499852 499854 499856 499858 499860 499862 499864 499866 499868 499870 499
872 499874 499876 499878 499880 499882 499884 499886 499888 499890 499892 499894
499896 499898 499899 499900 499901 499902 499903 499904 499905 499906 499907 499
918 499920 499922 499924 499926 499928 499930 499932 499934 499936 499938 4999
499942 499944 499946 499948 499950 499952 499954 499956 499958 499960 499962 49
9964 499966 499968 499970 499972 499974 499976 499978 499980 499982 499984 49998
86 499988 499990 499992 499994 499996 499998 499999 499998 499998 499998 499998
Clock ticks = 4960 .000000
oem@AllHaider: ~
```

T = 20 , N = 300

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 #define N = 300000;
10 int T = 20;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17 int WL = (int) (N/T);
18 int SID = TID * WL;
19
20 for (int l = SID ; l < SID+WL ; l++){
21     vector_res[l] = vector_1[l] + vector_2[l];
22 }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int l = 0 ; l < N; l++){
35         vector_1[l] = l;
36         vector_2[l] = l;
37     }
38
39     double start = clock();
40
41     for(int l = 0 ; l < T ; l++){
42         int tid = (int) malloc(sizeof(int));
43         tid = l;
44         pthread_create(&t1,NULL,vector_Sum,tid);
45     }
46
47     for(int l = 0 ; l < T ; l++){
48 }
```

```
6 599508 599510 599512 599514 599516 599518 599520 599522 599524 599526 599528
599530 599532 599534 599536 599538 599540 599542 599544 599546 599548 599550 599
52 599554 599556 599558 599560 599562 599564 599566 599568 599570 599572 599574
599576 599578 599580 599582 599584 599586 599588 599590 599592 599594 599596 599
58 599600 599602 599604 599606 599608 599610 599612 599614 599616 599618 59962
0 599622 599624 599626 599628 599630 599632 599634 599636 599638 599640 599642 59
64 599644 599646 599648 599650 599652 599654 599656 599658 599660 599662 599664 599
66 599666 599668 599670 599672 599674 599676 599678 599680 599682 599684 599686 599
68 599688 599690 599692 599694 599696 599698 599700 599702 599704 599706 599708 599
70 599710 599712 599714 599716 599718 599720 599722 599724 599726 599728 599730 599732
599734 599736 599738 599740 599742 599744 599746 599748 599750 599752 599754 599756 59
78 599768 599770 599772 599774 599776 599778 599780 599782 599784 599786 599788 599790
0 599792 599794 599796 599798 599799 599800 599802 599803 599804 599805 599806 599807 59
9808 599810 599812 599814 599816 599818 599820 599822 599824 599826 599828 599830 599832
599834 599836 599838 599840 599842 599844 599846 599848 599850 599852 599854 599856 599858
599856 599852 599854 599856 599858 599860 599862 599864 599866 599868 599870 599
872 599874 599876 599878 599880 599882 599884 599886 599888 599890 599892 599894
599896 599898 599899 599900 599902 599904 599906 599908 599910 599912 599914 599916 59
918 599920 599922 599924 599926 599928 599930 599932 599934 599936 599938 599940 5999
42 599944 599946 599948 599950 599952 599954 599956 599958 599960 599962 599964 59996
66 599966 599968 599970 599972 599974 599976 599978 599980 599982 599984 599986 59998
86 599988 599990 599992 599994 599996 599998 599999 599998 599998 599998 599998
Clock ticks = 7901 .000000
oem@AllHaider: ~
```

T = 20 , N = 350

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Assignment2.c" and it contains C code for a vector sum operation. The output of the program is displayed in the terminal, consisting of a long sequence of numbers (clock ticks) from 6 to 99998. The desktop background is purple, and the taskbar at the bottom shows various application icons.

```
06 69958 699518 699512 699514 699516 699518 699528 699522 699524 699526 699528  
699530 699532 699534 699536 699538 699540 699542 699544 699546 699548 699550 699  
552 699554 699556 699558 699560 699562 699564 699566 699568 699570 699572 699574  
699576 699578 699580 699582 699584 699586 699588 699590 699592 699594 699596 69  
9598 699600 699602 699604 699606 699608 699610 699612 699614 699616 699618 699620  
0 699622 699624 699626 699628 699630 699632 699634 699636 699638 699640 699642 6  
99644 699646 699648 699650 699652 699654 699656 699658 699660 699662 699664 6996  
66 699668 699678 699692 699694 699696 699698 699700 699702 699704 699706 699708 699710 699  
712 699714 699716 699718 699720 699722 699724 699726 699728 699730 699732 699734  
699736 699738 699740 699742 699744 699746 699748 699750 699752 699754 699756 699758 699  
758 699760 699762 699764 699766 699768 699770 699772 699774 699776 699778 699780 699782  
0 699782 699784 699786 699788 699790 699792 699794 699796 699798 699800 699802 6  
99884 699886 699888 699890 699891 699892 699893 699894 699895 699896 699897 699898 699899  
26 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898  
699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898  
699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898  
699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898  
699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898 699898  
Clock ticks = 8702.000000  
oe@AllHolder:~
```

T = 20 , N = 400

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Assignment2.c" and it contains C code for a vector sum operation. The output of the program is displayed in the terminal, consisting of a long sequence of numbers (clock ticks) from 6 to 99998. The desktop background is purple, and the taskbar at the bottom shows various application icons.

```
06 79958 799518 799512 799514 799516 799518 799528 799522 799524 799526 799528  
799530 799532 799534 799536 799538 799540 799542 799544 799546 799548 799550 799  
552 799554 799556 799558 799560 799562 799564 799566 799568 799570 799572 799574  
799576 799578 799580 799582 799584 799586 799588 799590 799592 799594 799596 79  
9598 799600 799602 799604 799606 799608 799610 799612 799614 799616 799618 799620  
0 799622 799624 799626 799628 799630 799632 799634 799636 799638 799640 799642 6  
99644 799646 799648 799650 799652 799654 799656 799658 799660 799662 799664 7996  
66 799668 799678 799692 799694 799696 799698 799700 799702 799704 799706 799708 799710 799  
712 799714 799716 799718 799720 799722 799724 799726 799728 799730 799732 799734  
799736 799738 799740 799742 799744 799746 799748 799750 799752 799754 799756 799758 79  
9758 799760 799762 799764 799766 799768 799770 799772 799774 799776 799778 799780 799782  
0 799782 799784 799786 799788 799790 799792 799794 799796 799798 799800 799802 7  
99884 799886 799888 799890 799891 799892 799893 799894 799895 799896 799897 799898 799898  
26 799898 799898 799898 799898 799898 799898 799898 799898 799898 799898 799898 799898  
799898 799898 799898 799898 799898 799898 799898 799898 799898 799898 799898 799898  
799898 799898 799898 799898 799898 799898 799898 799898 799898 799898 799898 799898  
799898 799898 799898 799898 799898 799898 799898 799898 799898 799898 799898 799898  
Clock ticks = 6209.000000  
oe@AllHolder:~
```


T = 25 , N = 150

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Assignment2.c" and the command entered is "oemp@AllHaider:~\$". The output of the command is a very long string of numbers, likely a memory dump or a large file's contents. The desktop background is a purple gradient, and the taskbar at the bottom shows various application icons.

```
oemp@AllHaider:~$
```

```
6 299508 299510 299512 299514 299516 299518 299520 299522 299524 299526 299528  
299530 299532 299534 299536 299538 299540 299542 299544 299546 299548 299550 299  
552 299554 299556 299558 299560 299562 299564 299566 299568 299570 299572 299574  
299576 299578 299580 299582 299584 299586 299588 299590 299592 299594 299596 299  
9598 299600 299602 299604 299606 299608 299610 299612 299614 299616 299618 29962  
0 299620 299622 299624 299626 299628 299630 299632 299634 299636 299638 299640 299  
9644 299646 299648 299650 299652 299654 299656 299658 299660 299662 299664 2996  
66 299668 299670 299672 299674 299676 299678 299680 299682 299684 299686 299688  
299690 299692 299694 299696 299698 299700 299702 299704 299706 299708 299710 299  
712 299714 299716 299718 299720 299722 299724 299726 299728 299730 299732 299734  
299736 299738 299740 299742 299744 299746 299748 299750 299752 299754 299756 29  
9758 299759 299761 299763 299765 299767 299769 299771 299773 299775 299777 299779 299  
0 299782 299784 299786 299788 299790 299792 299794 299796 299798 299800 299802 2  
99884 299886 299888 299890 299891 299892 299893 299894 299895 299896 299897 299898 29989  
26 299898 299898 299898 299898 299898 299898 299898 299898 299898 299898 299898 29989  
299898 299898 299898 299898 299898 299898 299898 299898 299898 299898 299898 29989  
872 299898 299898 299898 299898 299898 299898 299898 299898 299898 299898 299898 29989  
299898 299898 299898 299898 299898 299898 299898 299898 299898 299898 299898 29989  
9918 299898 29992 29994 29996 29998 29990 29992 29993 29994 29995 29996 29997 29998 2999  
0 29994 29994 29994 29994 29994 29994 29995 29995 29995 29995 29995 29995 29996 29996 2999  
29996 29996 29996 29996 29996 29996 29996 29996 29996 29996 29996 29996 29996 29996 2999  
86 29998 29998 29998 29998 29998 29998 29998 29998 29998 29998 29998 29998 29998 29998  
Clock ticks = 4566 .000000  
oemp@AllHaider:~$
```

T = 25 , N = 200

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Assignment2.c" and the command entered is "oemp@AllHaider:~\$". The output of the command is a very long string of numbers, similar to the previous screenshot but longer. The desktop background is a purple gradient, and the taskbar at the bottom shows various application icons.

```
oemp@AllHaider:~$
```

```
06 399508 399510 399512 399514 399516 399518 399520 399522 399524 399526 399528  
399530 399532 399534 399536 399538 399540 399542 399544 399546 399548 399550 399  
552 399554 399556 399558 399560 399562 399564 399566 399568 399570 399572 399574  
399576 399578 399580 399582 399584 399586 399588 399590 399592 399594 399596 39  
9598 399600 399602 399604 399606 399608 399610 399612 399614 399616 399618 39962  
0 399622 399624 399626 399628 399630 399632 399634 399636 399638 399640 399642 3  
99644 399646 399648 399650 399652 399654 399656 399658 399660 399662 399664 3996  
66 399668 399670 399672 399674 399676 399678 399680 399682 399684 399686 399688  
399690 399692 399694 399696 399698 399700 399702 399704 399706 399708 399710 399  
712 399714 399716 399718 399720 399722 399724 399726 399728 399730 399732 399734  
399736 399738 399740 399742 399744 399746 399748 399750 399752 399754 399756 39  
9758 399760 399762 399764 399766 399768 399770 399772 399774 399776 399778 39978  
0 399782 399784 399786 399788 399790 399792 399794 399796 399798 399800 399802 3  
99884 399886 399888 399890 399891 399892 399893 399894 399895 399896 399897 39989  
26 399898 399898 399898 399898 399898 399898 399898 399898 399898 399898 399898 39989  
872 399898 399898 399898 399898 399898 399898 399898 399898 399898 399898 399898 39989  
399898 399898 399898 399898 399898 399898 399898 399898 399898 399898 399898 39989  
9918 399898 39992 39994 39996 39998 39990 39992 39993 39994 39995 39996 39997 3999  
0 39994 39994 39994 39994 39994 39994 39995 39995 39995 39995 39995 39995 39996 3999  
29996 39996 39996 39996 39996 39996 39996 39996 39996 39996 39996 39996 39996 3999  
86 39998 39998 39998 39998 39998 39998 39998 39998 39998 39998 39998 39998 39998  
Clock ticks = 4077 .000000  
oemp@AllHaider:~$
```

T = 25 , N = 250

The terminal window displays a large sequence of numbers (clock ticks) from 06 to 999988, followed by the command "Clock ticks = 4196.000000". The code editor window shows a C++ program named "Assignment2.c" that performs a vector sum operation using multiple threads.

```
06 499508 499510 499512 499514 499516 499518 499520 499522 499524 499526 499528  
499530 499532 499534 499536 499538 499540 499542 499544 499546 499548 499550 499552  
499554 499556 499558 499560 499562 499564 499566 499568 499570 499572 499574  
499576 499578 499580 499582 499584 499586 499588 499590 499592 499594 499596 49  
9598 499600 499602 499604 499606 499608 499610 499612 499614 499616 499618 49962  
0 499622 499624 499626 499628 499630 499632 499634 499636 499638 499640 499642 4  
99644 499646 499648 499650 499652 499654 499656 499658 499660 499662 499664 499666  
499668 499670 499672 499674 499676 499678 499680 499682 499684 499686 499688 499690  
499692 499694 499696 499698 499700 499702 499704 499706 499708 499710 499712  
712 499714 499716 499718 499720 499722 499724 499726 499728 499730 499732 499734  
499736 499738 499740 499742 499744 499746 499748 499750 499752 499754 499756 49  
9758 499760 499762 499764 499766 499768 499770 499772 499774 499776 499778 499780  
0 499782 499784 499786 499788 499790 499792 499794 499796 499798 499800 499802 5  
99804 499806 499808 499810 499812 499814 499816 499818 499820 499822 499824 4998  
26 499828 499830 499832 499834 499836 499838 499840 499842 499844 499846 499848  
499856 499852 499854 499856 499858 499860 499862 499864 499866 499868 499870 4998  
872 499874 499876 499878 499880 499882 499884 499886 499888 499890 499892 499894  
499896 499898 499900 499902 499904 499906 499908 499910 499912 499914 499916 499  
9918 499920 499922 499924 499926 499928 499930 499932 499934 499936 499938 49994  
0 499942 499944 499946 499948 499950 499952 499954 499956 499958 499960 499962 4  
99964 499966 499968 499970 499972 499974 499976 499978 499980 499982 499984 49998  
86 499988 499990 499992 499994 499996 499998 499999 499998 499992 499994 499996 499998  
Clock ticks = 4196.000000  
oem@AllHalder: ~
```

T = 25 , N = 300

The terminal window displays a large sequence of numbers (clock ticks) from 06 to 999988, followed by the command "Clock ticks = 7588.000000". The code editor window shows a C++ program named "Assignment2.c" that performs a vector sum operation using multiple threads.

```
06 599508 599510 599512 599514 599516 599518 599520 599522 599524 599526 599528  
599530 599532 599534 599536 599538 599540 599542 599544 599546 599548 599550 599552  
599554 599556 599558 599560 599562 599564 599566 599568 599570 599572 599574  
599576 599578 599580 599582 599584 599586 599588 599590 599592 599594 599596 59  
9598 599600 599602 599604 599606 599608 599610 599612 599614 599616 599618 59962  
0 599622 599624 599626 599628 599630 599632 599634 599636 599638 599640 599642 5  
99644 599646 599648 599650 599652 599654 599656 599658 599660 599662 599664 599666 5996  
68 599668 599670 599672 599674 599676 599678 599680 599682 599684 599686 599688 599690 599692  
599694 599696 599698 599700 599702 599704 599706 599708 599710 599712 599714 599  
712 599714 599716 599718 599720 599722 599724 599726 599728 599730 599732 599734  
599736 599738 599740 599742 599744 599746 599748 599750 599752 599754 599756 59  
9758 599760 599762 599764 599766 599768 599770 599772 599774 599776 599778 599780  
0 599782 599784 599786 599788 599790 599792 599794 599796 599798 599800 599802 5  
99804 599806 599808 599810 599812 599814 599816 599818 599820 599822 599824 599826 5998  
28 599828 599830 599832 599834 599836 599838 599840 599842 599844 599846 599848  
599850 599852 599854 599856 599858 599860 599862 599864 599866 599868 599870 599  
872 599874 599876 599878 599880 599882 599884 599886 599888 599890 599892 599894  
599896 599898 599900 599902 599904 599906 599908 599910 599912 599914 599916 59  
9918 599920 599922 599924 599926 599928 599930 599932 599934 599936 599938 59994  
0 599942 599944 599946 599948 599950 599952 599954 599956 599958 599960 599962 5  
99964 599966 599968 599970 599972 599974 599976 599978 599980 599982 599984 59998  
86 599988 599990 599992 599994 599996 599998 599999 599998 599996 599994 599992 59999  
Clock ticks = 7588.000000  
oem@AllHalder: ~
```

T = 25 , N = 350

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 350000;
10 int T = 25;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_sum(int TID){
17     int WL = (int)(N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&t1,NULL,vector_sum,tid);
46     }
47     for(int i = 0 ; i < T ; i++) {
```

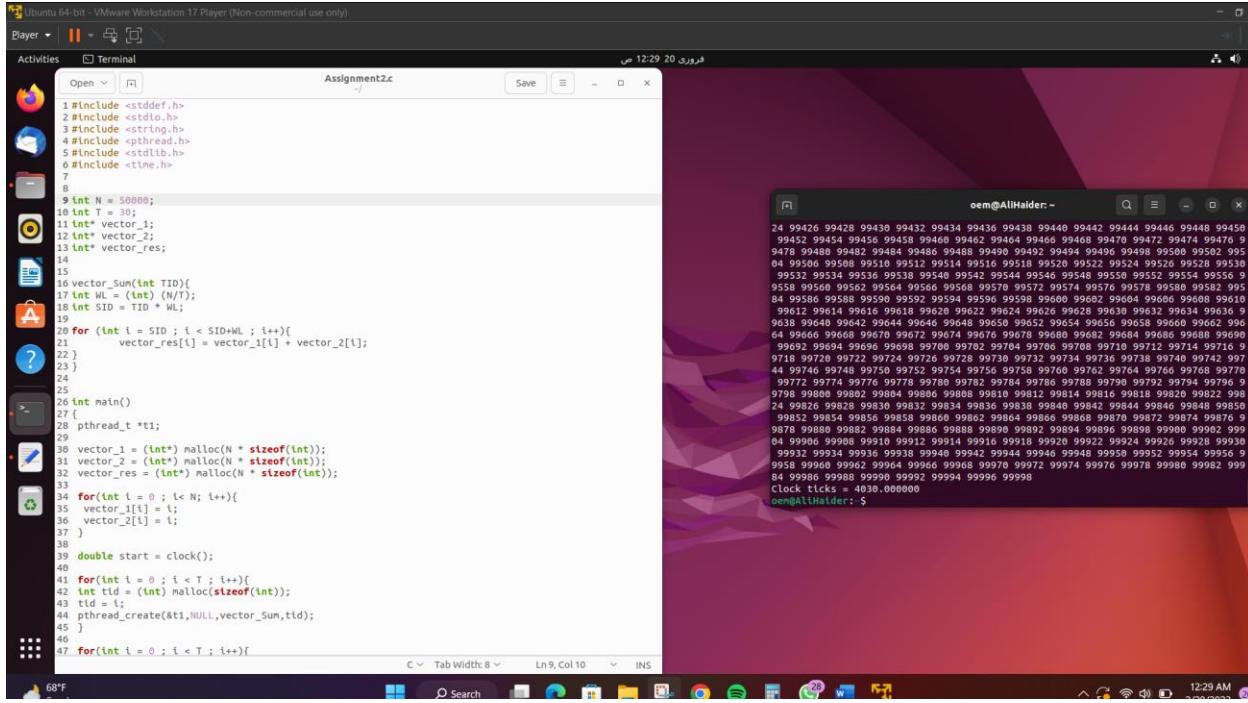
```
06 699508 699518 699512 699514 699516 699518 699520 699522 699524 699526 699528
699530 699532 699534 699536 699538 699540 699542 699544 699546 699548 699550 699
552 699552 699554 699556 699558 699560 699562 699564 699566 699568 699570 699571 69957
799576 699578 699580 699582 699584 699586 699588 699590 699592 699594 699596 699598 69959
9598 699600 699602 699604 699606 699608 699610 699612 699614 699616 699618 69962
0 699622 699624 699626 699628 699630 699632 699634 699636 699640 699642 6
99644 699646 699648 699650 699652 699654 699656 699658 699660 699662 699664 6996
66 699668 699670 699672 699674 699676 699678 699680 699682 699684 699686 699688 699690
699692 699694 699696 699698 699699 699700 699702 699704 699706 699708 699710 699
718 699712 699714 699716 699718 699720 699722 699724 699726 699728 699730 699732 69973
4 699736 699738 699740 699742 699744 699746 699748 699750 699752 699754 699756 699
9758 699760 699762 699764 699766 699768 699770 699772 699774 699776 699778 69978
6 699782 699784 699786 699788 699790 699792 699794 699796 699798 699800 699802 6
99804 699806 699808 699810 699812 699814 699816 699818 699820 699822 699824 699826 6998
28 699830 699832 699834 699836 699838 699840 699842 699844 699846 699848 699850 699852 69985
699854 699856 699858 699860 699862 699864 699866 699868 699870 699872 699874 699876 69987
872 699878 699880 699882 699884 699886 699888 699890 699892 699894 699896 699898 6998
699896 699898 699900 699902 699904 699906 699908 699910 699912 699914 699916 699
9918 699920 699922 699924 699926 699928 699930 699932 699934 699936 699938 69994
0 699942 699944 699946 699948 699950 699952 699954 699956 699958 699960 699962 6
99964 699966 699968 699970 699972 699974 699976 699978 699980 699982 699984 699986 69998
86 699988 699990 699992 699994 699996 699998 699999 699998 699982 699984 699986 699988
Clock ticks = 7196.000000
oem@AllHalider: ~
```

T = 25 , N = 400

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 400000;
10 int T = 25;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_sum(int TID){
17     int WL = (int)(N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&t1,NULL,vector_sum,tid);
46     }
47     for(int i = 0 ; i < T ; i++) {
```

```
06 799508 799510 799512 799514 799516 799518 799520 799522 799524 799526 799528
799530 799532 799534 799536 799538 799540 799542 799544 799546 799548 799550 799
552 799552 799554 799556 799558 799560 799562 799564 799566 799568 799570 799572 79957
799576 799578 799580 799582 799584 799586 799588 799590 799592 799594 799596 799598 79959
9598 799600 799602 799604 799606 799608 799610 799612 799614 799616 799618 79962
0 799622 799624 799626 799628 799630 799632 799634 799636 799638 799640 799642 7
99644 799646 799648 799650 799652 799654 799656 799658 799660 799662 799664 79966
66 799668 799670 799672 799674 799676 799678 799680 799682 799684 799686 799688 79968
799690 799692 799694 799696 799698 799700 799702 799704 799706 799708 799710 799
712 799714 799716 799718 799720 799722 799724 799726 799728 799730 799732 799734
799736 799738 799740 799742 799744 799746 799748 799750 799752 799754 799756 79
9758 799760 799762 799764 799766 799768 799770 799772 799774 799776 799778 79978
0 799782 799784 799786 799788 799790 799792 799794 799796 799798 799790 799792 79979
9804 799806 799808 799810 799812 799814 799816 799818 799820 799822 799824 799826 799828 7998
26 799828 799830 799832 799834 799836 799838 799840 799842 799844 799846 799848 79984
799850 799852 799854 799856 799858 799860 799862 799864 799866 799868 799870 799
782 799874 799876 799878 799880 799882 799884 799886 799888 799890 799892 799894 7998
799896 799898 799900 799902 799904 799906 799908 799910 799912 799914 799916 799
9918 799920 799922 799924 799926 799928 799930 799932 799934 799936 799938 79994
0 799942 799944 799946 799948 799950 799952 799954 799956 799958 799960 799962 7
999964 799966 799968 799970 799972 799974 799976 799978 799980 799982 799984 79998
86 799988 799990 799992 799994 799996 799998 799999 799998 799982 799984 799986 79998
Clock ticks = 8196.000000
oem@AllHalider: ~
```

T = 30 , N = 50



```

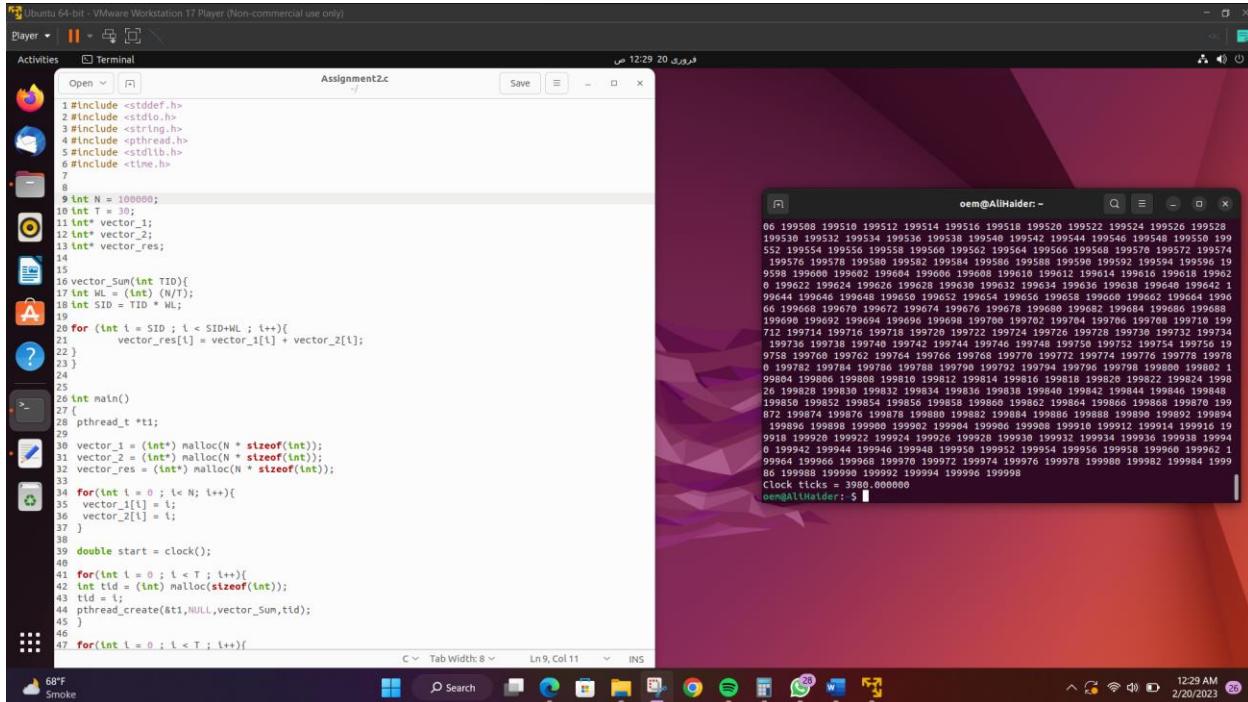
Ubuntu 64-bit - VMware Workstation 17 Player (Non-commercial use only)
Player Terminal فوری 20 12:29 ص
Activities Assignment2.c Save
Open [ ] [-] x
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 50000;
10 int T = 30;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int i = 0 ; i < T ; i++){
42         int tid = (int) malloc(sizeof(int));
43         tid = i;
44         pthread_create(&t1,NULL,vector_Sum,tid);
45     }
46
47     for(int i = 0 ; i < T ; i++){

```

```

oem@AllHaider:~ oem@AllHaider:-
24 99426 99428 99430 99432 99434 99436 99438 99440 99442 99444 99446 99448 99449 99450 99452 99454 99456 99458 99460 99462 99464 99466 99468 99470 99472 99474 99476 9
9478 99480 99482 99484 99486 99488 99490 99492 99494 99496 99498 99500 99502 99504 99506 99508 99510 99512 99514 99516 99518 99520 99522 99524 99526 99528 99530
99532 99534 99536 99538 99540 99542 99544 99546 99548 99550 99552 99554 99556 99558 99560 99562 99564 99566 99568 99570 99572 99574 99576 99578 9958 99582 995
9558 99584 99586 99588 99590 99592 99594 99596 99598 99600 99602 99604 99606 99608 99608 99610 99612 99614 99616 99618 99620 99622 99624 99626 99628 99630 99632 99634 99636 99638 99640 99642 99644 99646 99648 99650 99652 99654 99656 99658 99660 99662 996
64 99666 99668 99670 99672 99674 99676 99678 99680 99682 99684 99686 99688 99690 99692 99694 99696 99698 99700 99702 99704 99706 99708 99710 99712 99714 99716 9
99692 99694 99696 99698 99700 99702 99704 99706 99708 99710 99712 99714 99716 9
9718 99720 99722 99724 99726 99728 99730 99732 99734 99738 99740 99742 99744 99746 99748 99750 99752 99754 99756 99758 99760 99762 99764 99766 99768 99770 99772 99774 99776 99778
9978 99790 99792 99794 99796 99798 99800 99802 99804 99806 99808 99810 99812 99814 99816 99818 99820 99822 998
24 99826 99828 99830 99832 99834 99836 99838 99840 99842 99844 99846 99848 99850 99852 99854 99856 99858 99860 99862 99864 99866 99868 99870 99872 99874 99876 9
9878 99880 99882 99884 99886 99888 99890 99892 99894 99896 99898 99900 99902 99904 99906 99908 99910 99912 99914 99916 99918 99920 99922 99924 99926 99928 99930
04 99990 99992 99994 99996 99998 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999
99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999
84 99996 99998 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999
Clock ticks = 4030.000000
oem@AllHaider:~
```

T = 30 , N = 100



```

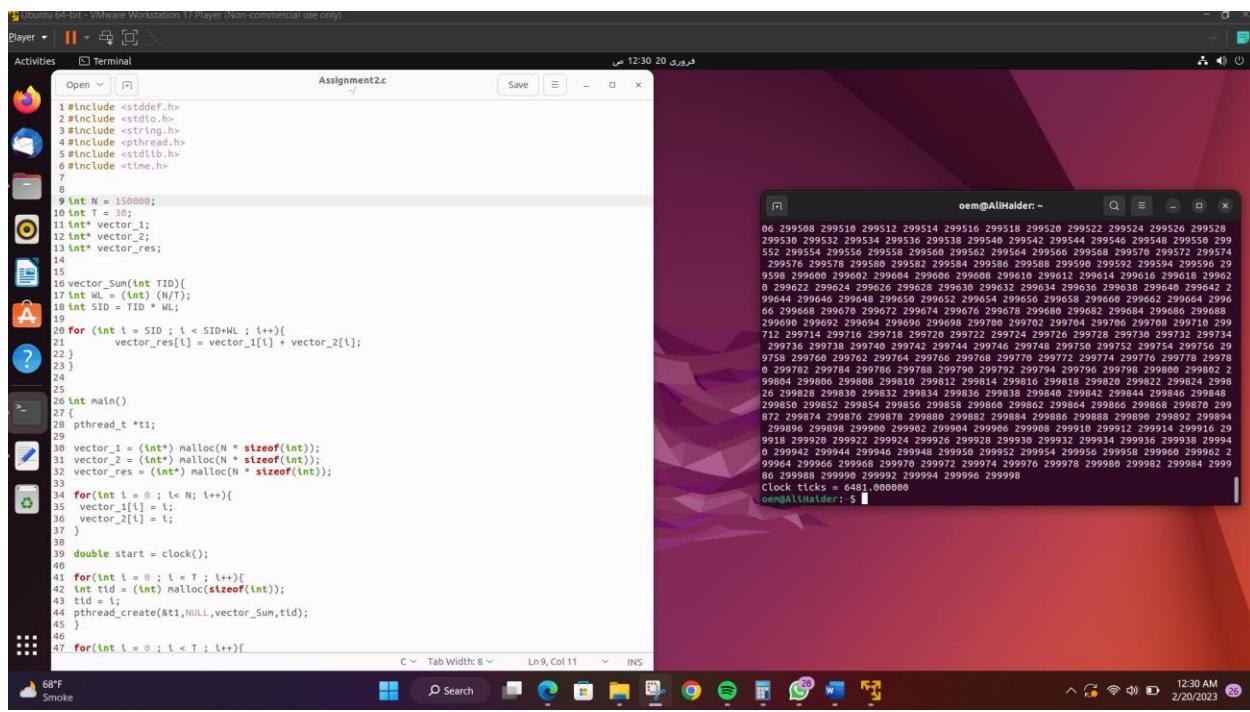
Ubuntu 64-bit - VMware Workstation 17 Player (Non-commercial use only)
Player Terminal فوری 20 12:29 ص
Activities Assignment2.c Save
Open [ ] [-] x
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 100000;
10 int T = 30;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26 int main()
27 {
28     pthread_t *t1;
29
30     vector_1 = (int*) malloc(N * sizeof(int));
31     vector_2 = (int*) malloc(N * sizeof(int));
32     vector_res = (int*) malloc(N * sizeof(int));
33
34     for(int i = 0 ; i < N; i++){
35         vector_1[i] = i;
36         vector_2[i] = i;
37     }
38
39     double start = clock();
40
41     for(int i = 0 ; i < T ; i++){
42         int tid = (int) malloc(sizeof(int));
43         tid = i;
44         pthread_create(&t1,NULL,vector_Sum,tid);
45     }
46
47     for(int i = 0 ; i < T ; i++){

```

```

oem@AllHaider:~ oem@AllHaider:-
06 199508 199510 199512 199514 199516 199518 199520 199522 199524 199526 199528
199530 199532 199534 199536 199538 199540 199542 199544 199546 199548 199550 199
552 199554 199556 199558 199560 199562 199564 199566 199568 199570 199572 199574 199576 199578 199580 199582 199584 199586 199588 199590 199592 199594 199596 199598 1995
9598 199600 199602 199604 199606 199608 199610 199612 199614 199616 199618 19962
0 199622 199624 199626 199628 199630 199632 199634 199636 199638 199640 199642 1
99644 199646 199648 199650 199652 199654 199656 199658 199660 199662 199664 19966
66 199668 199670 199672 199674 199676 199678 199680 199682 199684 199686 199688 199690 199692 199694 199696 199698 199700 199702 199704 199706 199708 199710 199
71 199712 199714 199716 199718 199720 199722 199724 199726 199728 199730 199732 199734 199736 199738 199740 199742 199744 199746 199748 199750 199752 199754 199756 19
9758 199768 199769 199770 199772 199774 199776 199778 199780 199782 199784 199786 199788 199790 199792 199794 199796 199798 199800 199802 1
99804 199806 199808 199810 199812 199814 199816 199818 199820 199822 199824 199826
26 199828 199830 199832 199834 199836 199838 199840 199842 199844 199846 199848 199850 199852 199854 199856 199858 199860 199862 199864 199866 199868 199870 199872 199874 199876 199878 199880 199882 199884 199886 199888 199890 199892 199894 199896 199898 199900 199902 199904 199906 199908 199910 199912 199914 199916 19
9918 199920 199922 199924 199926 199928 199930 199932 199934 199936 199938 19994
0 199942 199944 199946 199948 199950 199952 199954 199956 199958 199960 199962 199964 199966 199968 199970 199972 199974 199976 199978 199980 199982 199984 199986 199988 199990 199992 199994 199996 199998 19999
88 199998 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999 199999
Clock ticks = 3980.000000
oem@AllHaider:~
```

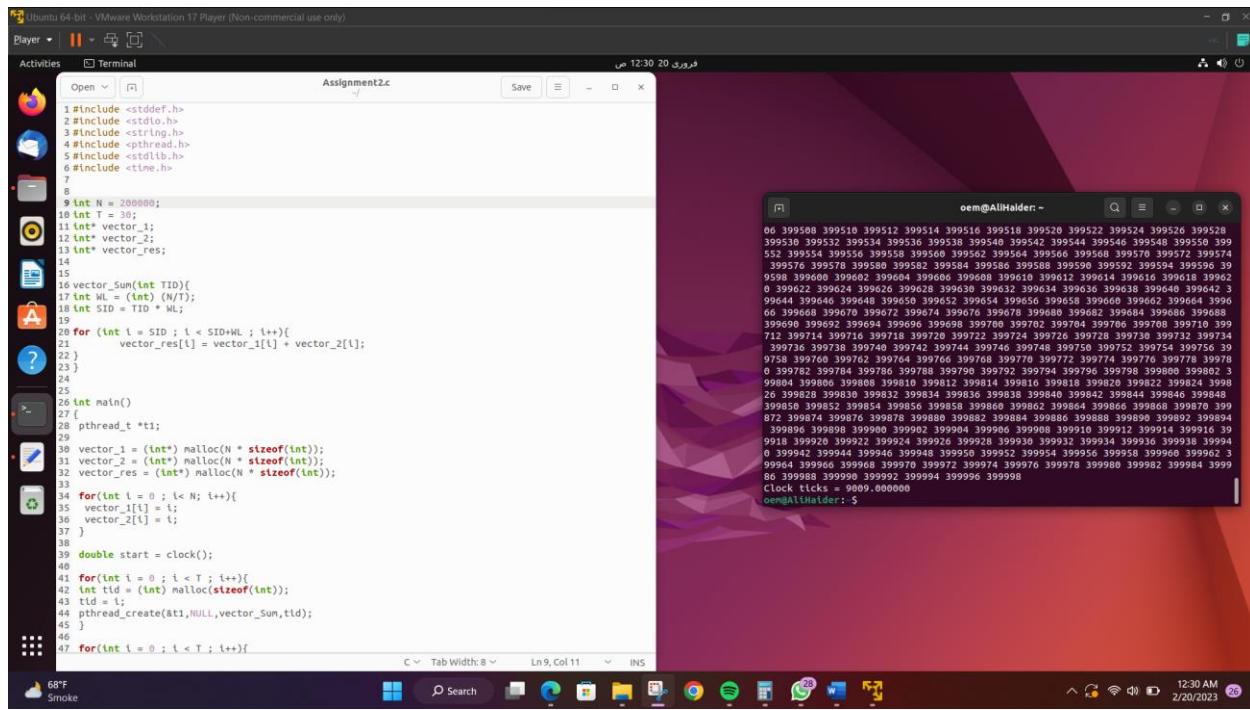
T = 30 , N = 150



The terminal window displays a large sequence of numerical values, likely the results of a computation. The values are arranged in several columns and rows, starting with 6, 299508, 299510, etc., and continuing through many more digits. The terminal window has a title bar "Assignment2.c" and a status bar at the bottom indicating "Clock ticks = 6481.000000". The desktop environment includes a taskbar with various icons like a browser, file manager, and system tray.

```
6 299508 299510 299512 299514 299516 299518 299520 299522 299524 299526 299528  
299530 299532 299534 299536 299538 299540 299542 299544 299546 299548 299550 299552 299554 299556 299558 299560 299562 299564 299566 299568 299570 299572 299574 299576 299578 299580 299582 299584 299586 299588 299590 299592 299594 299596 299598 299600 299602 299604 299606 299608 299610 299612 299614 299616 299618 299620 299622 299624 299626 299628 299630 299632 299634 299636 299638 299640 299642 299644 299646 299648 299650 299652 299654 299656 299658 299660 299662 299664 299666 299668 299670 299672 299674 299676 299678 299680 299682 299684 299686 299688 299690 299692 299694 299696 299698 299700 299702 299704 299706 299708 299710 299712 299714 299716 299718 299720 299722 299724 299726 299728 299730 299732 299734 299736 299738 299740 299742 299744 299746 299748 299750 299752 299754 299756 299758 299760 299762 299764 299766 299768 299770 299772 299774 299776 299778 299780 299782 299784 299786 299788 299790 299792 299794 299796 299798 299800 299802 299804 299806 299808 299810 299812 299814 299816 299818 299820 299822 299824 299826 299828 299830 299832 299834 299836 299838 299840 299842 299844 299846 299848 299850 299852 299854 299856 299858 299860 299862 299864 299866 299868 299870 299872 299874 299876 299878 299880 299882 299884 299886 299888 299890 299892 299894 299896 299898 299900 299902 299904 299906 299908 299910 299912 299914 299916 299918 299920 299922 299924 299926 299928 299930 299932 299934 299936 299938 299940 299942 299944 299946 299948 299950 299952 299954 299956 299958 299960 299962 299964 299966 299968 299970 299972 299974 299976 299978 299980 299982 299984 299986 299988 299990 299992 299994 299996 299998 299999 299999  
Clock ticks = 6481.000000  
os@AllHaider: ~
```

T = 30 , N = 200



The terminal window displays a large sequence of numerical values, similar to the previous screenshot but for a larger value of N (200). The values are arranged in several columns and rows, starting with 6, 399508, 399510, etc., and continuing through many more digits. The terminal window has a title bar "Assignment2.c" and a status bar at the bottom indicating "Clock ticks = 9009.000000". The desktop environment includes a taskbar with various icons like a browser, file manager, and system tray.

```
6 399508 399510 399512 399514 399516 399518 399520 399522 399524 399526 399528  
399530 399532 399534 399536 399538 399540 399542 399544 399546 399548 399550 399552 399554 399556 399558 399560 399562 399564 399566 399568 399570 399572 399574 399576 399578 399580 399582 399584 399586 399588 399590 399592 399594 399596 399598 399599 399600 399602 399604 399606 399608 399610 399612 399614 399616 399618 399620 399622 399624 399626 399628 399630 399632 399634 399636 399638 399640 399642 399644 399646 399648 399650 399652 399654 399656 399658 399660 399662 399664 399666 399668 399670 399672 399674 399676 399678 399680 399682 399684 399686 399688 399690 399692 399694 399696 399698 399700 399702 399704 399706 399708 399710 399712 399714 399716 399718 399720 399722 399724 399726 399728 399730 399732 399734 399736 399738 399740 399742 399744 399746 399748 399750 399752 399754 399756 399758 399760 399762 399764 399766 399768 399770 399772 399774 399776 399778 399780 399782 399784 399786 399788 399790 399792 399794 399796 399798 399800 399802 399804 399806 399808 399810 399812 399814 399816 399818 399820 399822 399824 399826 399828 399830 399832 399834 399836 399838 399840 399842 399844 399846 399848 399850 399852 399854 399856 399858 399860 399862 399864 399866 399868 399870 399872 399874 399876 399878 399880 399882 399884 399886 399888 399890 399892 399894 399896 399898 399899 399900 399902 399904 399906 399908 399910 399912 399914 399916 399918 399920 399922 399924 399926 399928 399930 399932 399934 399936 399938 399940 399942 399944 399946 399948 399950 399952 399954 399956 399958 399960 399962 399964 399966 399968 399970 399972 399974 399976 399978 399980 399982 399984 399986 399988 399989 399990 399992 399994 399996 399998 399999  
Clock ticks = 9009.000000  
os@AllHaider: ~
```

T = 30 , N = 250

The terminal window displays a large list of floating-point numbers ranging from 499508 to 99984, likely representing the results of a parallel computation. The code editor window shows a C++ program named 'Assignment2.c' that performs vector summation using multiple threads.

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8 int N = 250000;
9 int T = 30;
10 int* vector_1;
11 int* vector_2;
12 int* vector_res;
13
14
15
16 vector_Sum(int TID){
17     int WL = (int) (N/T);
18     int SID = TID * WL;
19
20     for (int i = SID ; i < SID+WL ; i++){
21         vector_res[i] = vector_1[i] + vector_2[i];
22     }
23 }
24
25
26
27 int main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&t1,NULL,vector_Sum,tid);
46     }
47
48     for(int i = 0 ; i < T ; i++){
49
50
51
52
53
54
55
56
57
58
59
599508 599510 599512 599514 599516 599518 599520 599522 599524 599526 599528
599530 599532 599534 599536 599538 599540 599542 599544 599546 599548 599550 599
552 599554 599556 599558 599560 599562 599564 599566 599568 599570 599572 599574 59957
599576 599578 599580 599582 599584 599586 599588 599590 599592 599594 599596 599598 59959
599599 599600 599602 599604 599606 599608 599610 599612 599614 599616 599618 59962
0 599620 599622 599624 599626 599628 599630 599632 599634 599636 599638 599640 599642 59964
599644 599646 599648 599650 599652 599654 599656 599658 599660 599662 599664 59966
599666 599668 599670 599672 599674 599676 599678 599680 599682 599684 599686 599688
599690 599692 599694 599696 599698 599700 599702 599704 599706 599708 599710 59971
72 599714 599716 599718 599720 599722 599724 599726 599728 599730 599732 599734 59973
499736 599738 599740 599742 599744 599746 599748 599750 599752 599754 599756 59975
599758 599760 599762 599764 599766 599768 599770 599772 599774 599776 599778 59977
0 599782 599784 599786 599788 599790 599792 599794 599796 599798 599800 599802 59980
599804 599806 599808 599810 599812 599814 599816 599818 599820 599822 599824 5998
26 599828 599830 599832 599834 599836 599838 599840 599842 599844 599846 599848 59984
499850 599852 599854 599856 599858 599860 599862 599864 599866 599868 599870 59987
2 599874 599876 599878 599880 599882 599884 599886 599888 599890 599892 599894 59989
6 599896 599898 599900 599902 599904 599906 599908 599910 599912 599914 599916 599
918 499920 499922 499924 499926 499930 499932 499934 499936 499938 499940 499942 499944
499946 499948 499950 499952 499954 499956 499958 499960 499962 499964 499966 499968
499970 499972 499974 499976 499978 499980 499982 499984 499986 499988 499990 499992
86 499998 599999 599992 599994 599996 599998 599999 599999 599999 599999 599999 59999
8 Clock ticks = 10534.000000
oem@AliHaider: ~
```

T = 30 , N = 300

The terminal window displays a large list of floating-point numbers ranging from 599508 to 599984, similar to the previous run but with different values. The code editor window shows the same C++ program for vector summation.

```
1 #include <stddef.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <pthread.h>
5 #include <stdlib.h>
6 #include <time.h>
7
8
9 int N = 300000;
10 int T = 30;
11 int* vector_1;
12 int* vector_2;
13 int* vector_res;
14
15
16
17 vector_Sum(int TID){
18     int WL = (int) (N/T);
19     int SID = TID * WL;
20
21     for (int i = SID ; i < SID+WL ; i++){
22         vector_res[i] = vector_1[i] + vector_2[i];
23     }
24 }
25
26
27 int main()
28 {
29     pthread_t *t1;
30
31     vector_1 = (int*) malloc(N * sizeof(int));
32     vector_2 = (int*) malloc(N * sizeof(int));
33     vector_res = (int*) malloc(N * sizeof(int));
34
35     for(int i = 0 ; i< N; i++){
36         vector_1[i] = i;
37         vector_2[i] = i;
38     }
39
40     double start = clock();
41
42     for(int i = 0 ; i < T ; i++){
43         int tid = (int) malloc(sizeof(int));
44         tid = i;
45         pthread_create(&t1,NULL,vector_Sum,tid);
46     }
47
48     for(int i = 0 ; i < T ; i++){
49
50
51
52
53
54
55
56
57
58
59
599508 599510 599512 599514 599516 599518 599520 599522 599524 599526 599528
599530 599532 599534 599536 599538 599540 599542 599544 599546 599548 599550 599
552 599554 599556 599558 599560 599562 599564 599566 599568 599570 599572 59957
599576 599578 599580 599582 599584 599586 599588 599590 599592 599594 599596 59959
599598 599600 599602 599604 599606 599608 599610 599612 599614 599616 599618 59962
0 599620 599622 599624 599626 599628 599630 599632 599634 599636 599638 599640 599642 59964
599644 599646 599648 599650 599652 599654 599656 599658 599660 599662 599664 59966
599666 599668 599670 599672 599674 599676 599678 599680 599682 599684 599686 599688
599690 599692 599694 599696 599698 599700 599702 599704 599706 599708 599710 59971
72 599714 599716 599718 599720 599722 599724 599726 599728 599730 599732 599734 59973
499736 599738 599740 599742 599744 599746 599748 599750 599752 599754 599756 59975
599758 599760 599762 599764 599766 599768 599770 599772 599774 599776 599778 59977
0 599782 599784 599786 599788 599790 599792 599794 599796 599798 599800 599802 59980
599804 599806 599808 599810 599812 599814 599816 599818 599820 599822 599824 5998
26 599828 599830 599832 599834 599836 599838 599840 599842 599844 599846 599848 59984
499850 599852 599854 599856 599858 599860 599862 599864 599866 599868 599870 59987
2 599874 599876 599878 599880 599882 599884 599886 599888 599890 599892 599894 59989
6 599896 599898 599900 599902 599904 599906 599908 599910 599912 599914 599916 599
918 499920 499922 499924 499926 499930 499932 499934 499936 499938 499940 499942 49994
499946 499948 499950 499952 499954 499956 499958 499960 499962 499964 499966 499968
499970 499972 499974 499976 499978 499980 499982 499984 499986 499988 499990 499992
86 499998 599999 599992 599994 599996 599998 599999 599999 599999 599999 599999 59999
8 Clock ticks = 8823.000000
oem@AliHaider: ~
```

T = 30 , N = 350

The terminal window displays a sequence of random integers ranging from 699508 to 999846, separated by spaces. The code editor shows a C program named `Assignment2.c` that performs a vector sum operation using multiple threads.

```
699508 699518 699512 699514 699516 699518 699522 699522 699524 699526 699528  
699530 699532 699534 699536 699538 699540 699542 699544 699546 699548 699550 699  
552 699554 699556 699558 699560 699562 699564 699566 699568 699570 699572 699574  
699576 699578 699580 699582 699584 699586 699588 699590 699592 699594 699596 69  
9598 699600 699602 699604 699606 699608 699610 699612 699614 699616 699618 6996  
0 699620 699622 699624 699626 699628 699630 699632 699634 699636 699638 699640 69964  
99644 699646 699648 699650 699652 699654 699656 699658 699660 699662 699664 6996  
66 699668 699670 699672 699674 699676 699678 699680 699682 699684 699686 699688  
699690 699692 699694 699696 699698 699700 699702 699704 699706 699708 699710 699  
712 699714 699716 699718 699720 699722 699724 699726 699728 699730 699732 699734  
699736 699738 699740 699742 699744 699746 699748 699750 699752 699754 699756 69  
9758 699759 699761 699763 699765 699767 699769 699771 699773 699775 699777 699779  
699782 699784 699786 699788 699790 699792 699794 699796 699798 699800 699802 6  
99884 699886 699888 699890 699892 699894 699896 699898 699900 699902 699904 699906 699908 699910 699912 699914 699916 699918 699920 699922 699924 699926 699928 699930 699932 699934 699936 699938 69994  
0 699942 699944 699946 699948 699950 699952 699954 699956 699958 699960 699962 699964 699966 699968 699970 699972 699974 699976 699978 699980 699982 699984 699986 699988 699990 699992 699994 699996 699998 699990 699992 699994 699996 699998  
Clock ticks = 9652.000000  
oem@AliHaider:~
```

T = 30 , N = 400

The terminal window displays a sequence of random integers ranging from 799508 to 799846, separated by spaces. The code editor shows the same C program as the previous screenshot, but with a larger value of N (400) and a higher value of T (30).

```
699508 799510 799512 799514 799516 799518 799522 799524 799526 799528  
799530 799532 799534 799536 799538 799540 799542 799544 799546 799548 799550 799  
552 799554 799556 799558 799560 799562 799564 799566 799568 799570 799572 799574  
799576 799578 799580 799582 799584 799586 799588 799590 799592 799594 799596 79  
9598 799600 799602 799604 799606 799608 799610 799612 799614 799616 799618 7996  
0 799620 799622 799624 799626 799628 799630 799632 799634 799636 799638 799640 79964  
2 799644 799646 799648 799650 799652 799654 799656 799658 799660 799662 799664 7996  
6 799668 799670 799672 799674 799676 799678 799680 799682 799684 799686 799688  
799690 799692 799694 799696 799698 799700 799702 799704 799706 799708 799710 799  
712 799714 799716 799718 799720 799722 799724 799726 799728 799730 799732 799734  
799736 799738 799740 799742 799744 799746 799748 799750 799752 799754 799756 79  
9758 799759 799761 799763 799765 799767 799769 799771 799773 799775 799777 799779  
799782 799784 799786 799788 799790 799792 799794 799796 799798 799800 799802 7  
99884 799886 799888 799890 799892 799894 799896 799898 799900 799902 799904 799906 799908 799910 799912 799914 799916 799918 799920 799922 799924 799926 799928 799930 799932 799934 799936 799938 79994  
0 799942 799944 799946 799948 799950 799952 799954 799956 799958 799960 799962 799964 799966 799968 799970 799972 799974 799976 799978 799980 799982 799984 799986 799988 799990 799992 799994 799996 799998  
Clock ticks = 8540.000000  
oem@AliHaider:~
```

Prepared By
Muhammad Ali Haider
23047