Hello world

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

int main (int argc, char *argv[]) {
  int th_id, nthreads;
  #pragma omp parallel private(th_id)
  {
    th_id = omp_get_thread_num();
    printf("Hello World from thread %d\n", th_id);
    #pragma omp barrier
    if ( th_id == 0 ) {
      nthreads = omp_get_num_threads();
      printf("There are %d threads\n",nthreads);
    }
  }
  return 0;
}
```

Parallel for

```
int main(int argc, char **argv) {
    const int N = 100000;
    int i, a[N];

    #pragma omp parallel for
    for (i = 0; i < N; i++)
        a[i] = 2 * i;

    return 0;
 }
```

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>

/*defines the total amount of iterations*/
#define N_ITER 1024
int main(void)
{
    int i,id,chunk_size,numit;
    int array[N_ITER];
    memset(array,0,sizeof(array));
#pragma omp parallel default(none) private(i, id, numit) shared(array)
    {
numit=0;
id = omp_get_thread_num();
printf ("Thread no %d starting... \n", id);
srand(time(0)*(1+id));
#pragma omp for schedule(runtime) private(id)
for (i = 0; i < N_ITER; ++i)
{
    id = omp_get_thread_num();
    usleep(rand()%(10000*(1+id)));
    array[i]=id;
    numit++;
}
printf("Thread %d performed %d iterations\n",id,numit);
    }
    for (i = 0; i < N_ITER; ++i)
printf("%d",array[i]);
    puts("");
    return 0;
}
```

- $ env OMP_SCHEDULE="static,8" ./schedule
- Thread no 1 starting...
- Thread no 3 starting...
- Thread no 0 starting...
- Thread no 2 starting...
- Thread 3 performed 256 iterations
- Thread 0 performed 256 iterations
- Thread 2 performed 256 iterations
- Thread 1 performed 256 iterations
- 00000000111111112222222233333333000000001111111122222222333333330000000011111111
- 12222222333333330000000011111111222222223333333300000000111111112222222233333333
- 33000000001111111122222222333333330000000011111111222222223333333300000000111111
- 11122222222333333330000000111111112222222233333333000000001111111122222222333333
- 33330000000011111111222222223333333300000000111111112222222233333333000000001111
- 11111222222223333333300000000111111112222222233333333000000001111111122222222333
- 33333300000000111111112222222233333333000000001111111122222222333333330000000011
- 11111112222222233333333000000001111111122222222333333330000000011111111222222223
- 33333333000000001111111122222222333333330000000011111111222222223333333300000000
- 01111111122222222333333330000000011111111222222223333333300000000111111112222222
- 22333333330000000011111111222222223333333300000000111111112222222233333333000000
- 00011111111222222223333333300000000111111112222222233333333000000001111111122222
- 22223333333300000000111111112222222233333333000000001111111122222222333333333

- $ env OMP_SCHEDULE="dynamic,8" ./schedule
- Thread no 2 starting...
- Thread no 0 starting...
- Thread no 1 starting...
- Thread no 3 starting...
- Thread 3 performed 120 iterations
- Thread 0 performed 480 iterations
- Thread 1 performed 256 iterations
- Thread 2 performed 168 iterations
- 22222222000000001111111133333333000000001111111100000000000000002222222200000000
- 01111111333333330000000000000000111111112222222200000000000000001111111100000000
- 00222222220000000033333333111111110000000000000000222222220000000011111111000000
- 00033333333000000011111111000000002222222200000000111111113333333300000000111100
- 11112222222200000000111111110000000022222222000000003333333311111111000000002222
- 22222000000000000000111111110000000011111111000000001111111122222222333333330000
- 00000011111111000000000000000022222222111111110000000033333333000000001111111000
- 00000002222222200000000111111110000000022222222000000003333333311111111000000000
- 00000000000000001111111122222222000000001111111133333333000000002222222000000000
- 01111111100000000333333330000000011111111000000002222222200000000111111100000000
- 00333333330000000022222222111111110000000011111111000000002222222333333330000000
- 00000000001111111100000000222222220000000011111111000000002222222333333330000000
- 00001111111100000000222222221111111100000000333333330000000011111111000000000

- $ env OMP_SCHEDULE="guided,8" ./schedule
- Thread no 0 starting...
- Thread no 2 starting...
- Thread no 3 starting...
- Thread no 1 starting...
- Thread 2 performed 192 iterations
- Thread 0 performed 454 iterations
- Thread 1 performed 234 iterations
- Thread 3 performed 144 iterations
- 00000000000000000000000000000000000000000000000000000000000000000000000000000000
- 00000000000000000000000000000000000000000000000000000000000000000000000000000000
- 00000000000000000000000000000000000000000000000000000000000000000000000000000000
- 00000000000000000022222222222222222222222222222222222222222222222222222222222222
- 22222222222222222222222222222222222222222222222222222222222222222222222222222222
- 22222222222222222222222222222222222222222222222233333333333333333333333333333333
- 33333333333333333333333333333333333333333333333333333333333333333333333333333333
- 33333333333333333333333333333333331111111111111111111111111111111111111111111111
- 11111111111111111111111111111111111111111111111111111111111111111111111111111111
- 11111111111111111111111111111111111111111111111111111111111111111111111000000000
- 00000000000000000000000000000000000000000000000000000000000000000000000000000000
- 00000000000000000000000000000000000000000000000011111111111111111111111111111111
- 00000000000000000000000000000011111111110000000011111111000000000000000000000000

- $ env OMP_SCHEDULE="auto" ./schedule
- Thread no 0 starting...
- Thread no 1 starting...
- Thread no 3 starting...
- Thread no 2 starting...
- Thread 3 performed 256 iterations
- Thread 2 performed 256 iterations
- Thread 1 performed 256 iterations
- Thread 0 performed 256 iterations
- 00000000000000000000000000000000000000000000000000000000000000000000000000000000
- 00000000000000000000000000000000000000000000000000000000000000000000000000000000
- 00000000000000000000000000000000000000000000000000000000000000000000000000000000
- 00000000000000000011111111111111111111111111111111111111111111111111111111111111
- 11111111111111111111111111111111111111111111111111111111111111111111111111111111
- 11111111111111111111111111111111111111111111111111111111111111111111111111111111
- 11111111111111111111111111111111111112222222222222222222222222222222222222222222
- 22222222222222222222222222222222222222222222222222222222222222222222222222222222
- 22222222222222222222222222222222222222222222222222222222222222222222222222222222
- 22222222222222222222222222222222222222222222222222222233333333333333333333333333
- 33333333333333333333333333333333333333333333333333333333333333333333333333333333
- 33333333333333333333333333333333333333333333333333333333333333333333333333333333
- 33333333333333333333333333333333333333333333333333333333333333333333333333333333

```
#define N 10000 /*size of a*/
void calculate(int); /*The function that calculates the elements of a*/
int i;
long w;
long a[N];
calculate(a);
long sum = 0;
/*forks off the threads and starts the work-sharing construct*/
#pragma omp parallel for private(w) reduction(+:sum) schedule(static,1)
for(i = 0; i < N; i++)
   {
     w = i*i;
     sum = sum + w*a[i];
   }
printf("\n %li",sum);
```

```
...
long sum = 0, loc_sum = 0;
/*forks off the threads and starts the work-sharing construct*/
#pragma omp parallel for private(w,loc_sum) schedule(static,1)
{
  for(i = 0; i < N; i++)
    {
      w = i*i;
      loc_sum = loc_sum + w*a[i];
    }
  #pragma omp critical
  sum = sum + loc_sum;
}
printf("\n %li",sum);
```