

① 7ª Aula

Variáveis de ambiente

Definem o comportamento do código paralelo em tempo de execução.

→ `OMP_NUM_THREADS`: indica número de atividades que serão executadas em paralelo (default: número de processadores)

②

→ `OMP_SCHEDULE`: define esquema de escalonamento das atividades paralelas (default: static)

→ `OMP_DYNAMIC`: indica se o número de atividades a serem executadas em paralelo de ou não ser ajustado dinamicamente (default: FALSE)

→ `OMP_NESTED`: indica se deve ser contemplado ativação de paralelismo aninhado (default: FALSE)

③ Funções de interface

Conjunto de funções declaradas em `<omph>`

→ Controle do ambiente de execução

→ Sincronização ou bloqueio

→ Medição de tempo

④ Controle de ambiente de execução

→ `void omp_set_num_threads(int num)`

Define o número de threads padrão a serem usadas na região paralela

→ `int omp_get_num_threads()`

Retorna o número de threads ativas na região paralela onde foi chamada

→ `int omp_get_max_threads()`

Retorna o número de threads disponíveis para executar a região paralela

⑤

→ `int omp_get_thread_num()`

Retorna o identificador da thread relativo ao grupo ao qual pertence

→ `int omp_get_num_procs()`

Retorna o número de processadores disponíveis no momento da chamada da função

⑥ Sincronização ou bloqueio

→ Restringe a execução de um trecho de código a uma thread por vez

→ Permite maior flexibilidade que o uso de diretivas `critical` e `atomic`

→ Opera sobre variáveis "lock" do tipo `omp_lock_t` e `omp_nest_lock_t`

⑦

- `omp_init_lock()`
Inicializa um "lock"
- `omp_set_lock()`
Bloqueia a thread até que o "lock" especificado esteja disponível (desbloqueado)
- `omp_unset_lock()`
Desbloqueia um "lock"
- `omp_destroy_lock()`
Altera o estado de "lock" para não-inicializado

⑧ Exemplo

```
omp_lock_t *lock;  
omp_init_lock (&lock);  
  
#pragma omp parallel shared(lock, sum)  
{  
    #pragma omp for  
    for (i = 0; i < n; i++)  
    {  
        ...  
        omp_set_lock (&lock);  
        sum += i;  
        omp_unset_lock (&lock);  
    }  
}  
omp_destroy_lock (&lock);
```

⑨ Medição de tempo

double omp-get_wtime(void)

Retorna o valor decorrido em segundos relativo a um tempo passado

Exemplo:

```
double start, end;  
start = omp-get_wtime();  
...  
end = omp-get_wtime();  
printf("Tempo decorrido: %lf, end-start);
```

⑩ Atividades

(1) Integração numérica (Método dos trapézios)

Sequencial + Paralelo

(2) Multiplicação de matrizes

Sequencial + Paralelo