

頭の体操：以下の 1 から 100 の整数の和を取るプログラムを考える

```
Sum1to100()
{
int i;
int SumAll = 0,
for (i=1 ; i<=100; i++) {
    SumAll = SumAll + i;
}
}
```

これを 4 つの平行に動作する CPU で実行させようとして、以下の 4 つのプログラムを考えた

```
SumCPU0
{
int i;
int SumAll = 0;
int Sum1to25 = 0, Sum26to50 = 0, Sum51to75 = 0, Sum76to100 = 0;
for (i=1 ; i<=25; i++) {
    Sum1to25 = Sum1to25 + i;
}
SumAll = Sum1to25 + Sum26to50 + Sum51to75 + Sum76to100;
}
//
SumCPU1
{
int i;
int SumCPU1= 0;
for (i=26 ; i<=50; i++) {
    SumCPU1 = SumCPU1 + i;
}
Sum26to50 = SumCPU1;
}
//
SumCPU2
```

```
{
int i;
int SumCPU2= 0;
for (i=51 ; i<=75; i++) {
    SumCPU2 = SumCPU2 + i;
}
Sum51to75 = SumCPU2;
}
//
SumCPU3
{
int i;
int SumCPU3= 0;
for (i=76 ; i<=100; i++) {
    SumCPU3 = SumCPU3 + i;
}
Sum76to100 = SumCPU3;
}
```

○ 同期

## 9.1 背景

## 9.2 マルチコアの分類

### 図9. 4 共有メモリ型

#### 9.2.2 ヘテロ vs. ホモ

#### 9.3.1 共有メモリ方式と分散メモリ方式

- (1) 集中共有メモリ
- (2) 分散メモリ
- (3) 分散共有メモリ

#### 9.3.2 キャッシュ一貫性

- (1) スヌープ方式

#### 9.4.1 同期

- (1) 排他制御の必要性
- (2) スピンロック方式

#### 9.4.2 同期のためのハードウェア支援

- (1) テストアンドセット

#### 9.5 周辺機能

#### 図 9.30

#### 9.6.5 マルチコアで性能を気出すには

- (1) アムダールの法則

以上

以上