

# CISC 372: Parallel Computing

## OpenMP

Stephen F. Siegel

Department of Computer and Information Sciences  
University of Delaware

# OpenMP Overview



# OpenMP Overview

- ▶ an API for shared memory, multi-threaded programming
- ▶ works with C, C++, or Fortran







## Sequential Dot Product (Chapman et al., *Using OpenMP*)

```
#include<stdio.h>
int main() {
    double sum, a[256], b[256];
    int status, i, n=256;
    for (i = 0; i < n; i++) {
        a[i] = i * 0.5;
        b[i] = i * 2.0;
    }
    sum = 0;
    for (i = 0; i < n; i++) {
        sum = sum + a[i]*b[i];
    }
    printf("sum = %f \n", sum);
}
```



## Dot Product in OpenMP (Chapman et al., *Using OpenMP*)

```
#include<stdio.h>
int main() {
    double sum, a[256], b[256];
    int status, i, n=256;
    for (i = 0; i < n; i++) {
        a[i] = i * 0.5;
        b[i] = i * 2.0;
    }
    sum = 0;
    #pragma omp parallel for reduction(+:sum)
    for (i = 0; i < n; i++) {
        sum = sum + a[i]*b[i];
    }
    printf("sum = %f \n", sum);
}
```

# Basic Syntactic Concepts



## Basic Syntactic Concepts

- ▶ most directives are applied to the following **structured block** S
- ▶ S may be almost any kind of statement













## Basic Syntactic Concepts

- ▶ most directives are applied to the following **structured block** S
- ▶ S may be almost any kind of statement
  - ▶ a compound statement `{...}` (this is most common)
  - ▶ S must have single point of entry and single point of exit
  - ▶ may be a loop
    - ▶ if enclosing in curly braces would be a structured block
  - ▶ may be an *if* statement
    - ▶ if enclosing in curly braces would be a structured block
- ▶ in C, the directives have the form `#pragma omp ...`
- ▶ you can put non-newline white space before or after the `#`





## Spreading a logical line over multiple physical lines

```
#pragma omp this is my really big long \  
    pragma that keeps going and going and \  
    going on and on and on and on and on \  
    and on and on  
for (i=0; i<n; i++) {  
    ...  
}
```

**Beware:** You cannot have any white space after the `\`. It must be the last character on the physical line.





## Compiling and running an OpenMP program

- ▶ use gcc or clang, add flag `-fopenmp`; everything else the same

```
gcc -fopenmp -o dot dot.c
./dot
```
- ▶ without the flag `-fopenmp`
  - ▶ header file `omp.h` will not necessarily be found
  - ▶ pragmas will just be ignored; program will be sequential





























## hello1.c: parallel directive example

```
#include <stdio.h>

int main () {
    printf("I am the master.\n"); // just the master
#pragma omp parallel
    {
        printf("Hello, world.\n"); // all threads
    } /* end of parallel region */
    printf("Goodbye, world.\n"); // just the master
}
```











