

Parallel Computing

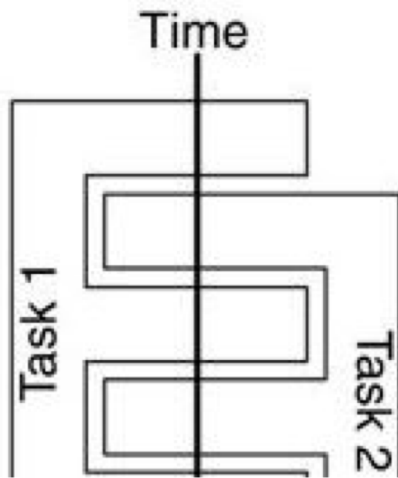


João Luís Ferreira Sobral
www.di.uminho.pt/~jls
jls@...

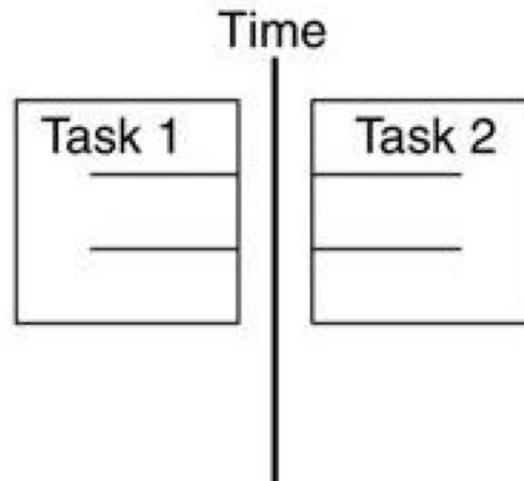
Web: Elearning

Logic vs physical parallelism

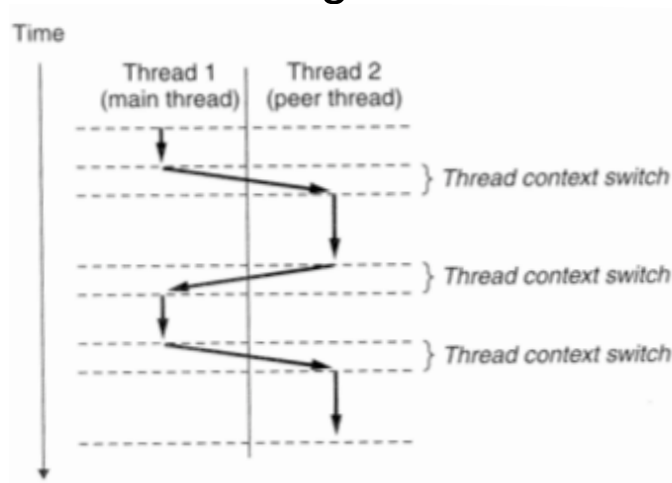
Concurrency



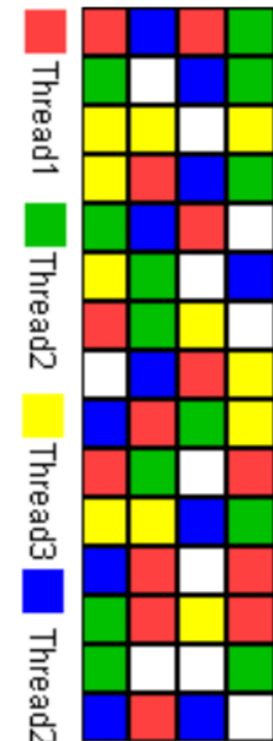
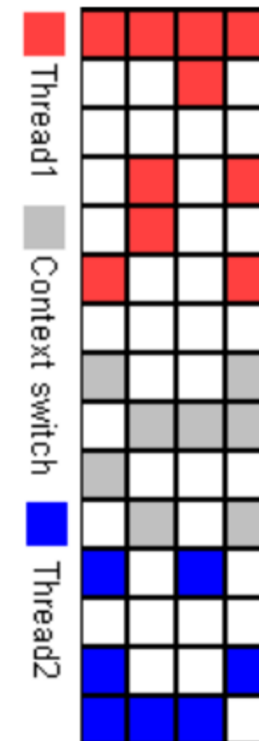
Parallelism



Scheduling



Intel SMT

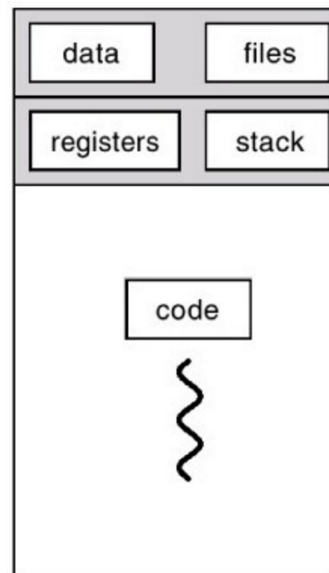


Specification of concurrency/parallelism

□ Processes

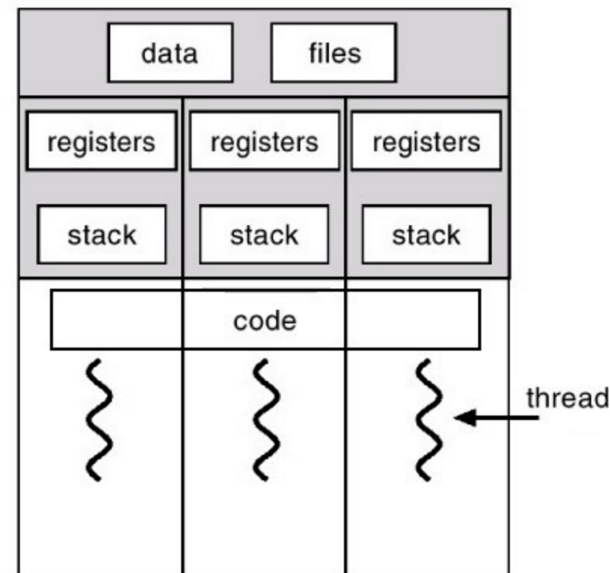
- Used for unrelated tasks
 - (e.g., a program)
- Own address space
 - Address space is protected from other process
- Switching at the kernel level

Every process has at least one thread

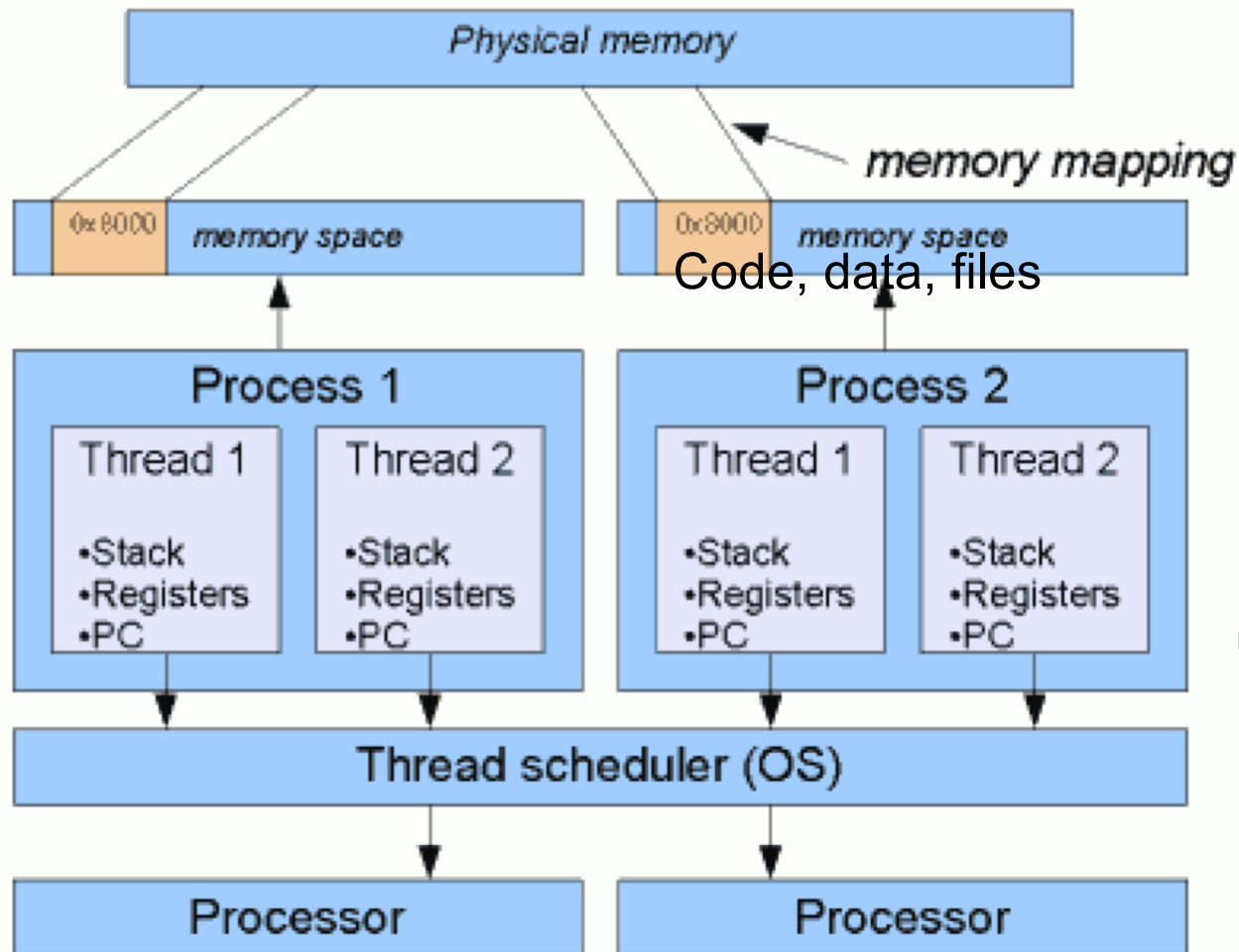


□ Threads

- Are part from the same job
- Share address space, code, data and files
- Switching at the user or kernel level



Thread vs Process



Data-races

< > data-race.c No Selection

```
1 #include <stdio.h>
2 #include <math.h>
3 #include <omp.h>
4
5 int main(){
6     double result={0};
7
8     #pragma omp parallel for shared(result)
9     for(int i=0; i<1000000;i++) {
10         result+=sin(i);
11     }
12     printf("%f",result);
13 }
```

vmovsd (%r12), %xmm1

L4:

```
vxorpd %xmm0, %xmm0, %xmm0
vcvttsi2sd %ebx, %xmm0, %xmm0
vmovsd %xmm1, 8(%rsp,%r12)
addl $1, %ebx ; i++
call _sin ; return value in %xmm0
vmovsd 8(%rsp,%r12), %xmm1 ; result in %xmm1
cmpl %ebx, %ebp ; i<1000000?
vaddsd %xmm0, %xmm1, %xmm1
jne L4
```

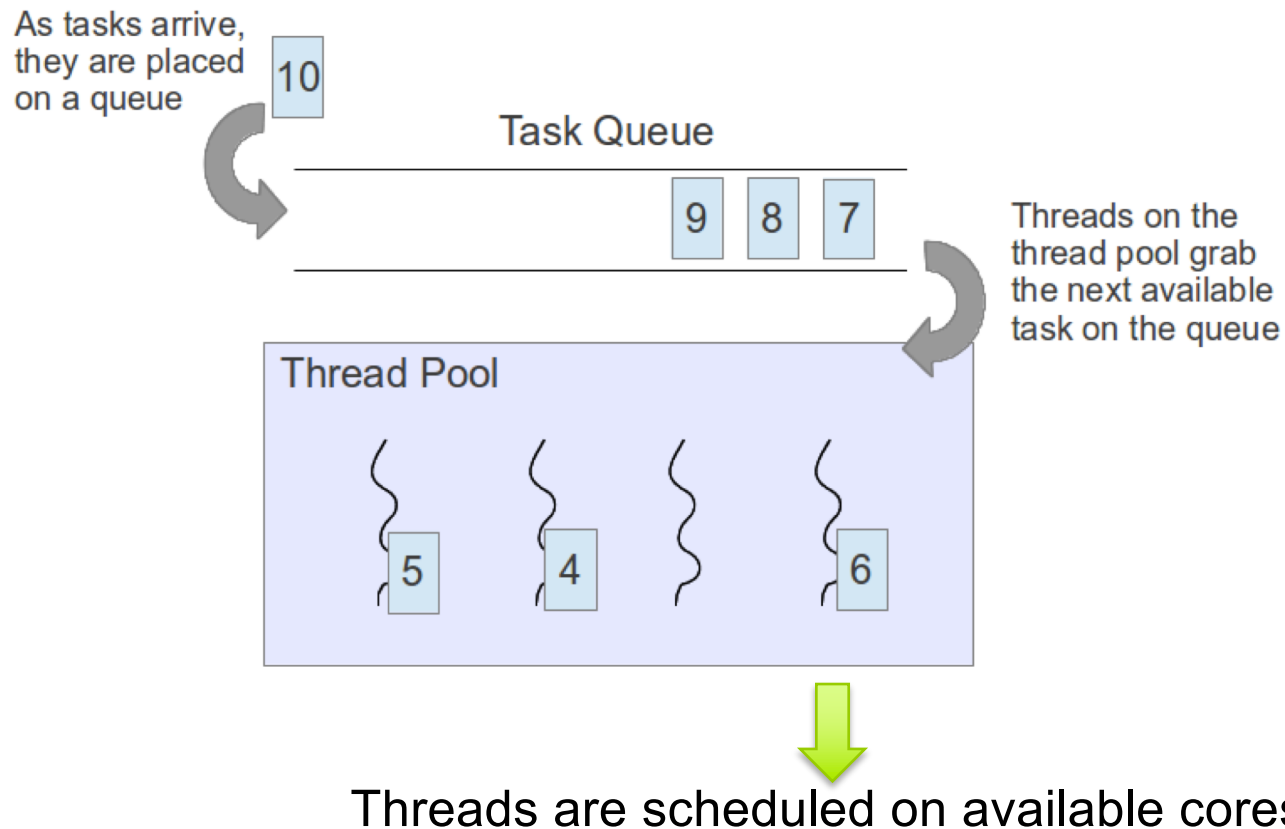
vmovsd %xmm1, (%r12)

Process/Thread vs Tasks

- **Task:** sequence of instructions
- **Thread/process:** execution context for a task
- **Processor/core:** hardware that runs a thread/process

In Java

- Runnable object
- Thread
- Processor core



Desenvolvimento de Aplicações Paralelas

Partição do problema e dos dados a processar

- Identifica **oportunidades de paralelismo**:
 - Define um elevado número de **tarefas** (de grão fino)
 - Pode obter várias decomposições alternativas
- Duas vertentes complementares na identificação das **tarefas**:
 - **Decomposição dos dados** - identifica dados que podem ser processados em paralelo
 - **enfoque nos dados** a processar e na sua divisão em conjuntos que podem ser processados em paralelo.
 - **Decomposição funcional** – identifica fases do algoritmo que podem ser efectuadas em paralelo.
 - **enfoque no processamento** a realizar, dividindo este processamento em tarefas independentes.
- A partição deve obter um número de **tarefas**, pelo menos, uma ordem de magnitude superior ao número de unidades de processamento
 - Introduce flexibilidade nas fases posteriores do desenvolvimento.
- Tarefas de dimensões idênticas facilitam a distribuição da carga
- O número de tarefas deve aumentar com a dimensão do problema.