Computação Paralela

Mest. Int. Engenharia Computacional
Mest. Engenharia Computacional

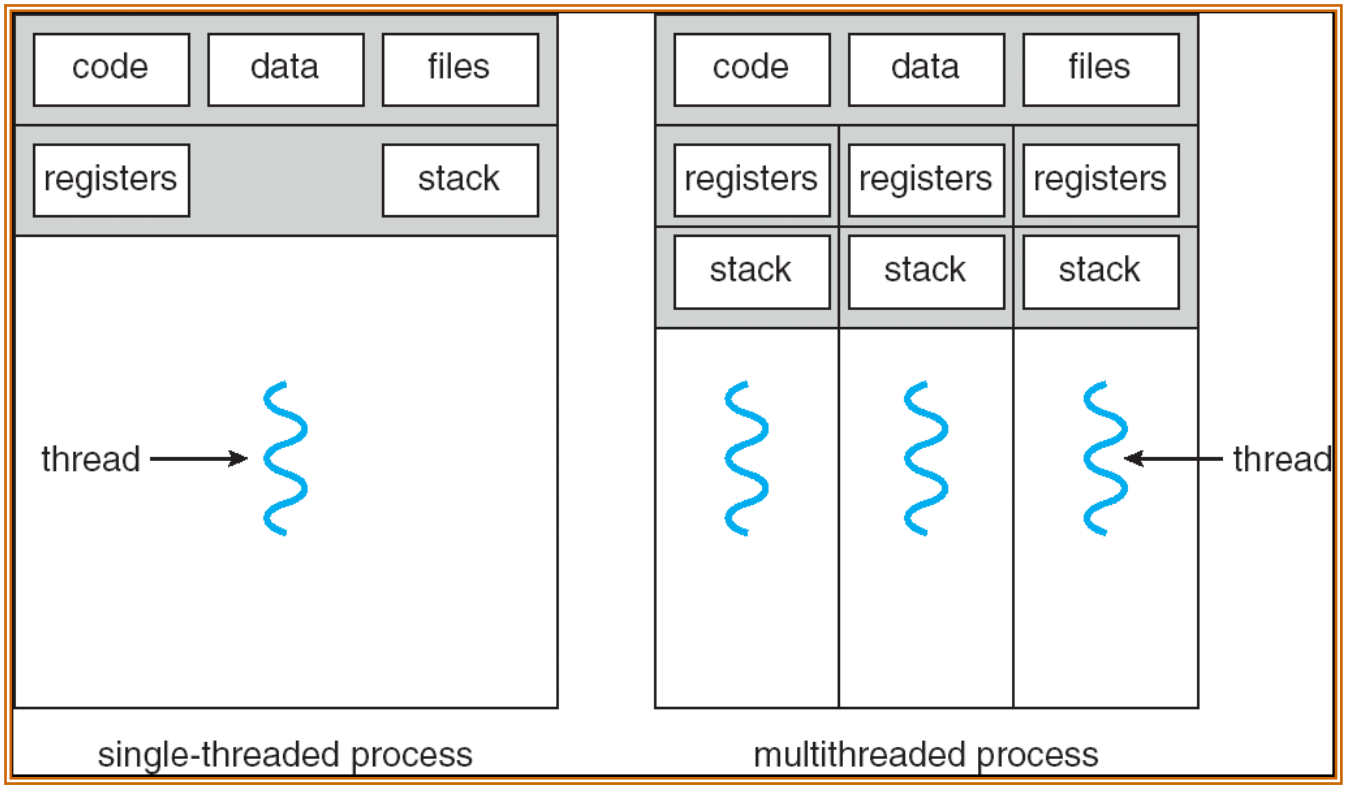
Ano letivo 2022/2023

Rui Costa, Nuno Lau

Threads

- Programas têm geralmente de executar diversas atividades distintas
- Usando *threads,* o programador pode desenvolver o programa como um conjunto de fluxos de execução sequenciais, um para cada atividade
- Cada *thread* comporta-se como tendo o seu processador próprio.
- Todas as *threads* do mesmo processo partilham espaço de endereçamento (memória)

Processos *Single* e *Multi threaded*



Processos e *Threads*

Per-process items	Per-thread items
Address space	Program counter
Global variables	Registers
Open files	Stack
Child processes	State
Pending alarms	
Signals and signal handlers	
Accounting information	

Servidor Web Multithreaded

- Num servidor web, cada pedido de página pode ser processado numa *thread* separada
- Há uma (*dispatcher*) *thread* que recebe todos os pedidos e os distribui pelas (*worker*) *threads*

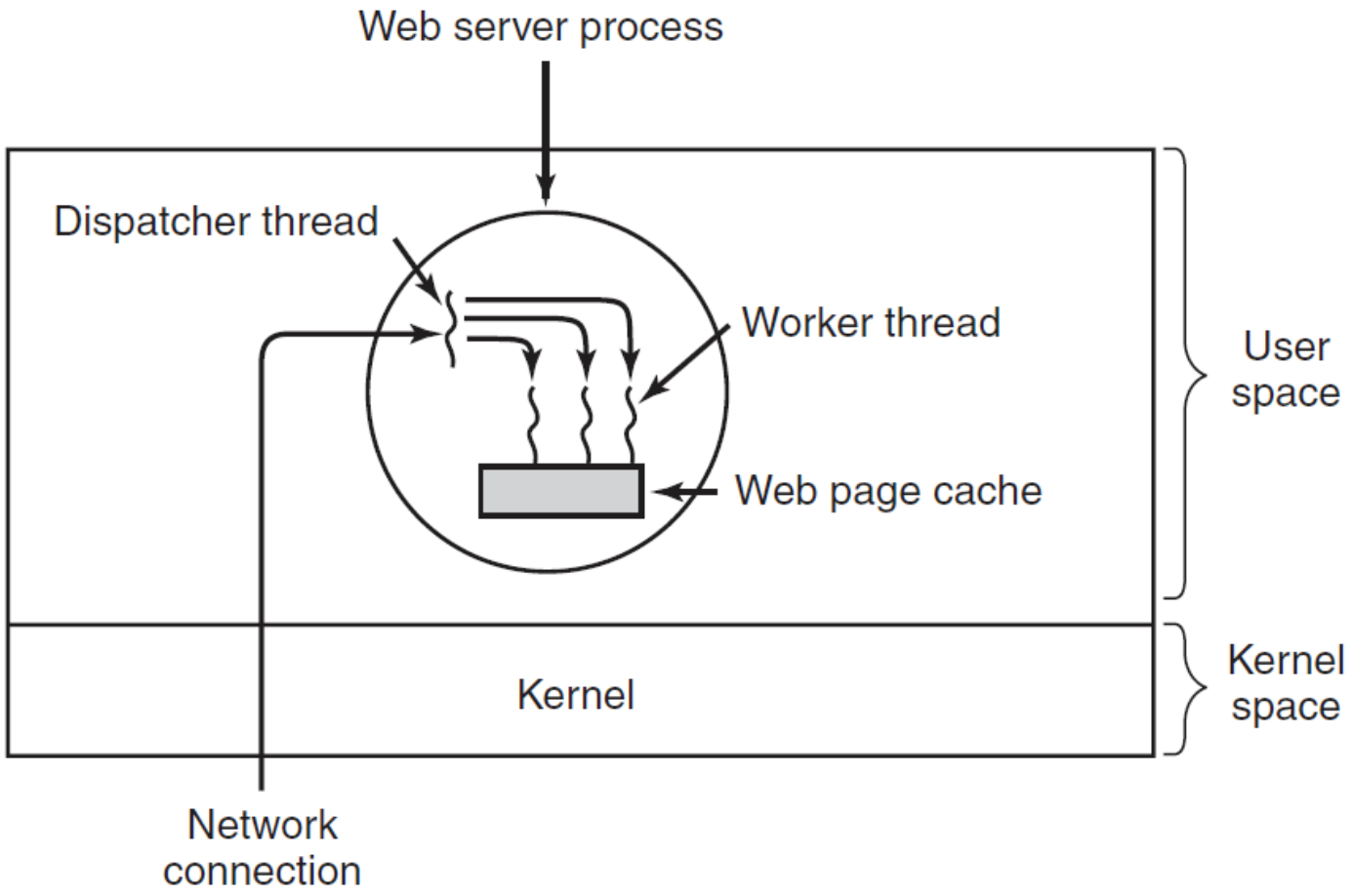
```
while (TRUE) {
    get_next_request(&buf);
    handoff_work(&buf);
}
```

Dispatcher thread

```
while (TRUE) {
    wait_for_work(&buf)
    look_for_page_in_cache(&buf, &page);
    if (page_not_in_cache(&page))
        read_page_from_disk(&buf, &page);
    return_page(&page);
}
```

Worker threads

Servidor Web Multithreaded



Pthreads

- POSIX standard para a criação e sincronização de *threads*
- API define comportamento, mas não implementação
- Comum em sistemas UNIX (Linux, Mac OS X)

POSIX *Threads*

universidade de aveiro
ua

Thread call	Description
Pthread_create	Create a new thread
Pthread_exit	Terminate the calling thread
Pthread_join	Wait for a specific thread to exit
Pthread_yield	Release the CPU to let another thread run
Pthread_attr_init	Create and initialize a thread's attribute structure
Pthread_attr_destroy	Remove a thread's attribute structure

Criar POSIX *Threads*

- **`int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine) (void *), void *arg);`**

```c
#include <stdio.h>
#include <pthread.h>

#define NUM_THREADS 5

void *PrintMsg(void *threadid) {
   long tid;
   tid = (long)threadid;
   printf("Hello World! Thread ID, %d\n", tid);
   pthread_exit(NULL);
}

int main (int argc, char *argv[]) {
   pthread_t threads[NUM_THREADS];
   int rc;
   int i;

   for( i = 0; i < NUM_THREADS; i++ ) {
      printf( "main() : creating thread, %d\n",i);
      rc = pthread_create(&threads[i], NULL, PrintMsg, (void *)i);

      if (rc) {
         printf("Error: unable to create thread, %d\n", rc);
         exit(1);
      }
   }
   pthread_exit(NULL);
}
```