

Processes and Threads

Networks and Embedded Software

Module 6.3

by Wolfgang Neff

Processes (1)

- Program vs Process
 - Program
 - Result of compilation
 - On hard disk
 - Does nothing
 - Static
 - Can be launched
 - Is a process then
 - Process
 - A running program
 - In main store
 - Does something
 - Dynamic
 - Has a context
 - Data
 - State (cf. 6.2.1)

Processes (2)

- Characteristics of Processes
 - Isolated from each other
 - Separate protection domain
 - No communication possible
 - Own copy of code, data and stack segment
 - Process control block on its own
 - Program counter
 - Registers
 - Memory, open files etc.

Processes (3)

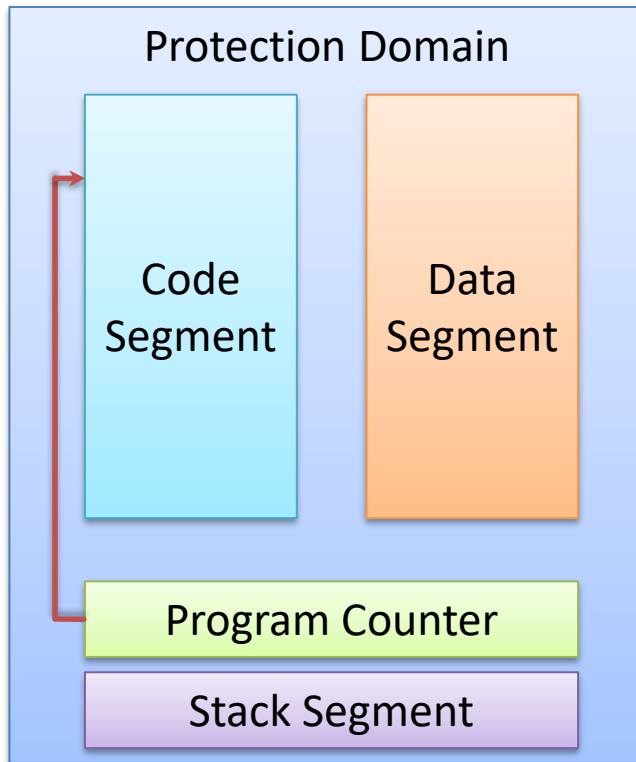
- Creation of a Process (Unix)
 - `#include <unistd.h>`
 - `pid = fork();`
 - `if (pid == 0) { // Child - New Process`
 - `// Do something or replace clone`
 - `// by another process`
 - `execl("AnotherProgramm", NULL);`
 - `}`
 - `else { // Parent - Old Process`
 - `// Do something`
 - `}`

Threads (1)

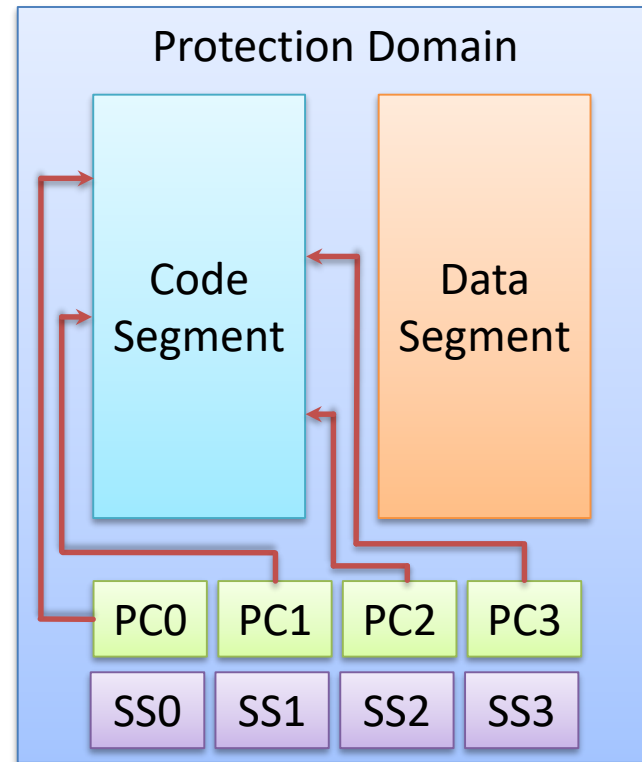
- Characteristics of Threads
 - Parallel task within a process
 - No protection at all
 - Easy communication by shared memory
 - Own copy of stack segment, only
 - Code and data segment are shared
 - Small thread control block on its own
 - Program counter, registers
 - Also known as lightweight processes

Threads (2)

- Process vs Thread



Single threaded Process



Process with 4 threads

Threads (3)

- Definition of the Thread's Task (POSIX-Thread)
 - `// Work load of the thread`
 - `void* task(void* arg);`
 - `// Do something`
 - `return NULL;`
 - `}`

Threads (4)

- Creation of a Thread (POSIX-Thread)
 - `#include <pthread.h>`
 - ...
 - `pthread_t thread;`
 - `pthread_create(&thread, NULL, task, NULL);`
 - `// Do something, thread runs parallel`
 - `pthread_join(thread, NULL);`
 - ...

Threads (5)

- Creation of a Critical Section (POSIX-Thread)
 - `#include <pthread.h>`
 - `pthread_mutex_t mutex;`
 - ...
 - `pthread_mutex_init(&mutex, NULL);`
 - ...
 - `pthread_mutex_lock(&mutex);`
 - `// Critical Section`
 - `pthread_mutex_unlock(&mutex);`
 - ...