

### 5.1. Procese si fire de executie (thread)

#### 5.1.1. Notiuni fundamentale

**Multitasking** – este o metoda prin care procese multiple (tasks) gestioneaza in comun resurse de procesare (timp procesor – CPU). In cazul sistemelor cu un singur procesor un singur proces poate fi rulat la un anumit moment de timp. Pentru rularea *simultana* a mai multor procese, metoda multitasking alocă si programează timpii CPU de executie ai fiecarui proces astfel incat pentru un interval de timp acestea sa ruleze virtual simultan. Procesul de realocare a resurselor CPU de la un proces la altul (context switch) va trebui sa se desfasoare cu o viteza suficient de mare pentru ca acest paralelism sa poate fi fructificat. Chiar si in sistemele multiprocesor, in general, sunt rulate mai multe procese decat numarul de unitati de procesare (CPU).

### 5.1. Procese si fire de executie (thread)

#### 5.1.1. Notiuni fundamentale

**Sistemele de operare** pot sa utilizeze una din urmatoarele trei tehnici de rulare a proceselor in sisteme multitasking:

- *Multiprogramming* – procesul curent este mentinut in executie pana cand o operatie cu un periferic necesita introducerea unui timp de asteptare sau sistemul de operare decide intreruperea executiei procesului. Aceste sisteme maximizeaza utilizarea resurselor CPU;
- *Time-sharing* – procesul curent elibereaza resursele ocupate fie voluntar, fie la aparitia unui eveniment extern de tip intrerupere. Intreruperea voluntara va aparea la intervale fixe de timp generate de un timer intern;
- *Real-time* – sisteme in care prioritizarea proceselor corelata cu un sistem de intreruperi ierarhizat determina executia si tratarea *simultana* a evenimentelor externe.



UNIA EUROPEANA



MINISTERUL EDUCATIEI SI  
CERCETARII



FONDUL SOCIAL EUROPEAN  
2007-2013



INSTRUMENTE STRUCTURALE  
2007-2013

### 5.1. Procese si fire de executie (thread)

#### 5.1.1. Notiuni fundamentale

##### Sisteme multiprogramming

Pentru primele sisteme de calcul costul timpului de procesare (CPU time) era ridicat si viteza perifericelor era scazuta. Astfel, de cele mai multe ori, accesarea unui periferic determina trecerea procesorului in stare de asteptare pana cand perifericul executa operatia curenta.

In 1960 s-au implementat primele sisteme multiprogram: cand programul curent ajungea la o operatie de citire a unui periferic, un alt program din lotul incarcat initial in memorie era rulat de CPU pana la finalizarea operatiei de accesare a perifericelor.

Dezavantaje:

- Variabilitatea timpului de executie al unui program



UNIA EUROPEANA



MINISTERUL EDUCATIEI SI  
CERCETARII



FONDUL SOCIAL EUROPEAN  
2007-2013



INSTRUMENTE STRUCTURALE  
2007-2013

### 5.1. Procese si fire de executie (thread)

#### 5.1.1. Notiuni fundamentale

##### **Sisteme time-sharing – multitasking co-operativ**

Pentru primele sisteme time-sharing erau constituite din suite de aplicatii care isi transferau voluntar timpii de acces la CPU prin corelarea co-operanta a nevoilor de procesare.

Ca urmare a faptului ca sistemele de tip co-operativ cedeaza in mod regulat accesul la CPU, proiectarea atenta a acestor SO trebuie efectuata in scopul evitarii posibilitatilor de blocare, mai ales in medii aleatoare (retele).

Exemple: Windows (pre 9x), MacOS (pre Osx), Windows 9x 16 bit legacy apps.



UNIUNEA EUROPEANĂ



MINISTERUL EDUCAȚIEI ȘI  
CERCETĂRII ȘTIINȚIFICE



FONDUL SOCIAL EUROPEAN  
2007-2013



INSTRUMENTE STRUCTURALE  
2007-2013

### 5.1. Procese si fire de executie (thread)

#### 5.1.1. Notiuni fundamentale

##### **Sisteme time-sharing – multitasking pre-emptiv**

Permit sistemelor embedded sa asigure fiecarui proces o “felie” corespunzatoare din timpul CPU si sa gestioneze rapid o cantitate mare de evenimente externe ce trebuie corelate cu un anumit proces.

In fiecare moment specific de executie procesele sunt grupate in doua categorii:

- Procese care sunt legate de opratii I/O (I/O bound);
- Procese legate strict de CPU (CPU bound);

Sistemul de operare aloca (poll, busywait) timpi de executie catre CPU bound in timp ce asteapta o anumita intrerupere externa. Sosirea datelor I/O determina realocarea timpilor CPU catre I/O bound. Aceste sisteme au fost imbunatatite de aparitia co-procesoarelor.

Exemple: Sinclair, Windows 9x, NT, MacOSx, Unix.

Mixed pre-emptive and co-operative.

### 5.1. Procese si fire de executie (thread)

#### 5.1.1. Notiuni fundamentale

##### Sisteme real-time

Sisteme in care prioritizarea proceselor corelata cu un sistem de intreruperi ierarhizat determina executia si tratarea *simultana* a evenimentelor externe. Aceste sisteme permit definirea unor procese cheie a caror executie este prioritara.

Termenul (real-time) deriva din simularea proceselor si utilizarea lui implica o viteza de calcul suficient de mare ca sa poata sa urmareasca viteza de desfasurare a evenimentelor din realitate.

Un sistem va fi denumit *real-time* daca rezultatul proceselor rulate va fi nu numai corect ci si disponibil in timp util:

- Hard real-time – rezultatul este inutil dupa expirarea timpului (automotive systems);
- Soft real-time – este permisa existenta unor timpi de latentă in detrimentul calitatii (digital video broadcasting);

### 5.1. Procese si fire de executie (thread)

#### 5.1.1. Notiuni fundamentale

##### Threads – fire de executie

Firele de executie au aparut din ideea ca cel mai eficient mod de cooperare intre procese este prin gestionarea comuna a intregului spatiu de memorie. Astfel, firele de executie sunt procese care ruleaza in acelasi context de memorie, schimbarea executiei de la un thread la altul neafectand contextul memoriei utilizate de CPU.

Firele de executie sunt programate pre-emptiv ca urmare a gestionarii comune a resurselor.

Fibrele (fibres) sunt segmente de proces asemanatoare firelor de executie insa pot fi programate co-operativ.

Pentru sisteme cu un singur CPU, firele de executie pot fi rulate simultan prin multiplexarea timpilor de executie cu o viteza suficient de mare – scheduler.



UNIA EUROPEANA



MINISTERUL EDUCATIEI SI  
CERCETARII



FONDUL SOCIAL EUROPEAN  
2007-2013



INSTRUMENTE STRUCTURALE  
2007-2013

### 5.1. Procese si fire de executie (thread)

#### 5.1.1. Notiuni fundamentale

##### Threads /procese

Diferente esentiale dintre aceste notiuni:

- Procesele sunt in general independente, firele sunt subseturi ale unui proces;
- Procesele contin informatii proprii de stare, firele au in comun aceleasi informatii de stare (memorie, resurse);
- Procesele au spatii de adresare separate, firele au acelasi spatiu de adresare;
- Procesele interactioneaza numai prin mecanisme proprii de inter-comunicare;
- Schimbarea contextului de executie intre fire este mai rapid decat intre procese.

Multithreading.



UNIUNEA EUROPEANĂ



MINISTERUL EDUCAȚIEI ȘI  
CERCETĂRII ȘTIINȚIFICE



FONDUL SOCIAL EUROPEAN  
2007-2013



INSTRUMENTE STRUCTURALE  
2007-2013