

# Operating System Process Schedulers

## Planificatoare de procese ale unui sistem de operare

### Tema 1

---

Responsabili: Alexandru Mustață și Ștefania Budulan

Data publicare temă: 25 oct. 2017

Soft Deadline: 14 nov. 2017, ora 10 PM

Hard Deadline: 21 nov. 2017, ora 10 PM (-10%/zi, până atunci)

### 1. Introducere

Scopul temei este acela de a vă familiariza cu programarea orientată pe obiecte și cu limbajul Java. În același timp, aveți ocazia să cunoașteți atât un mecanism folosit în practică, în kernelul oricărui sistem de operare - planificatorul de procese (va fi aprofundat în cadrul materiilor Sisteme de Operare I, II - SO, SO2) - , precum și un mecanism hardware care are rolul de a reduce diferența de viteză dintre processor și memoria RAM (conceptul de „memorie cache” va fi întâlnit în cadrul materiilor ED, CN1, CN2, SO, SO2, ASC).

### 2. Descrierea temei

Tema constă în implementarea mai multor planificatoare care vor programa task-uri (sarcini), respectând anumite politici. În momentul în care un task este programat (prin urmare, executat), el va prelua un număr de la input și va produce un șir de caractere (string) ca output.

#### 2.1 Descrierea tipurilor de task-uri

Există 7 tipuri de procese:

- **CheckPrime**
  - Verifică dacă numărul primit este prim.
  - Întoarce 0/1 (fals/adevărat).
- **NextPrime**
  - Returnează cel mai mic număr prim mai mare decât numărul primit.
- **Fibonacci**
  - Calculează al  $n$ -lea termen din șirul Fibonacci, dar *modulo (mod)* 9973, unde  $n$  este indexul primit. Dacă numărul (indexul) este negativ, întoarce -1.
  - e.g. pentru numărul 30 primit, va returna 4281.
    - cel de-al 30-lea termen din șirul Fibonacci este 832040;  $832040 \bmod 9973 = 4281$
- **Sqrt**
  - Returnează partea întreagă a radicalului modulului numărului primit.  
i.e. `[sqrt(|n|)]`
- **Square**

- Returnează pătratul numărului primit.
- **Cube**
  - Returnează cubul numărului primit.
- **Factorial**
  - Returnează factorialul mod 9973 al numărului primit. Dacă numărul este negativ, întoarce 0. i.e.  $n! \bmod 9973$

## 2.2 Descrierea tipurilor de planificatoare

- **RandomScheduler**

Selectează, aleator, un proces pentru rulare dintre procesele existente în sistem. Va fi utilizat pentru a verifica corectitudinea calculului care se efectuează de către fiecare proces.

- **RoundRobinScheduler**

Selectează un process astfel încât, în orice moment de timp, să existe o diferență de maximum o unitate între numărul de rulări ale oricaror două procese.

- **WeightedScheduler**

Definim cota unui proces ca fiind *procentul de rulări* ale procesului respectiv, raportat la rulările din întregul sistem.

Acest planificator va selecta un proces astfel încât, la momente de timp multiplu de  $t$ , unde

$$t = \text{suma}(\text{cote}) / \text{cmmdc}(\text{cote}),$$

să asigure respectarea cotelor impuse inițial de către administrator.

### Exemplu:

Notăție:  $\langle \text{Proces}, \text{Cota} \rangle$

Pentru mulțimea  $\{\langle \text{CheckPrime}, 1 \rangle, \langle \text{Sqrt}, 2 \rangle, \langle \text{Cube}, 4 \rangle\}$  înseamnă că, la momentele de timp multiplu de 7, rulările lui CheckPrime vor reprezenta 1/7 din toate rulările.

Pentru mulțimea  $\{\langle \text{Square}, 25 \rangle, \langle \text{Factorial}, 75 \rangle\}$  înseamnă că, la momentele de timp multiplu de 4, rulările lui Square vor reprezenta 1/4 din toate rulările.

## 2.3 Descrierea tipurilor de cache

- **NoCache**

Se rezolvă problema fără utilizarea unui mecanism de caching pentru rezultate (problema standard). Se va citi și numărul liniilor de cache, dar nu prezintă importanță aici.

- **LruCache (Least Recently Used) BONUS**

În momentul în care cache-ul se umple, se va șterge informația cea mai veche (de la cel mai mic moment de timp) pentru a face loc noii înregistrări.

- **LfuCache (Least Frequently Used) BONUS**

În momentul în care cache-ul se umple, se va șterge informația folosită de cele mai puține ori din cache pentru a face loc unui nou rezultat.

### 3. Descrierea inputului

Fișierul de input va avea urmatorul format:

```
CacheType
CacheLines
SchedulerType
NumberOfEvents
Process1 Weight1
...
LastProcess LastWeight
NumberOfNumbersToBeProcessed
Number1
...
LastNumber
```

Unde:

- *SchedulerType* este un String ce poate avea doar valorile: *RandomScheduler*, *RoundRobinScheduler* si *WeightedScheduler*.
- *NumberOfEvents* reprezintă numărul de evenimente ce există în sistem.
- *Process\_i* este un String ce reprezintă tipul procesului și poate avea doar valorile: *CheckPrime*, *NextPrime*, *Cube*, *Square*, *Sqrt*, *Factorial*, *Fibonacci*. Acesta are asociată o cotă, *Weight\_i*, care este un întreg pozitiv nenul.
- Numerele vor fi din intervalul [-1000, 1000] pentru a evita un rezultat de tip *overflow* pe tipul *int*, în cadrul anumitor operații.

### 4. Descrierea outputului

Fișierul de output va fi alcătuit din linii ce respectă următorul format:

```
<NumarulProcesat> <TipulProcesului> <Rezultatul> <FromCache/Computed>
```

### 5. Rularea temei

Tema va trebui să poată fi rulată astfel:

```
./tema1 <fisier_input> <fisier_output>
```

### 6. Punctare

- 30% Teste VMchecker RandomScheduler
- 30% Teste VMchecker RoundRobinScheduler
- 30% Teste VMchecker WeightedScheduler
- 10% Coding Style + Readme + JavaDoc + Makefile

- 20% Bonus Teste VMchecker LruCache & LfuCache

Notă: Temele care nu trec niciun test nu vor primi punctajul aferent *coding style și explicații* (de 10%).

## 7. Resurse

- [1] Round-robin scheduling – [https://en.wikipedia.org/wiki/Round-robin\\_scheduling](https://en.wikipedia.org/wiki/Round-robin_scheduling)
  - [2] Politici de înlocuire a liniilor de cache – [https://en.wikipedia.org/wiki/Cache\\_replacement\\_policies](https://en.wikipedia.org/wiki/Cache_replacement_policies)
- Planificare:
- [3] – [https://en.wikipedia.org/wiki/Scheduling\\_\(computing\)](https://en.wikipedia.org/wiki/Scheduling_(computing))
  - [4] – [https://www.tutorialspoint.com/operating\\_system/os\\_process\\_scheduling\\_algorithms.htm](https://www.tutorialspoint.com/operating_system/os_process_scheduling_algorithms.htm)
  - [5] – [https://www.tutorialspoint.com/operating\\_system/os\\_process\\_scheduling.htm](https://www.tutorialspoint.com/operating_system/os_process_scheduling.htm)
  - [6] Definirea comentariilor în cod – <http://www.oracle.com/technetwork/articles/java/index-137868.html>