

## Tema 2 – Structuri de date (seria CB)

### Switch-uri de rețea

Responsabili tema:	Andreea Stoican, Cosmin-Dumitru Oprea
Data publicarii:	<b>8.04.2016</b>
Termenul de predare:	<b>29.04.2016 ora 23:55</b> Se accepta teme trimise cu penalizare de 10 puncte / zi (din maxim 100 puncte) pana la data de 2.05.2016 ora 23:55

## 1. Introducere

Pentru a funcționa, o rețea locală de calculatoare are nevoie de dispozitive de interconectare. Un astfel de dispozitiv este switch-ul.



Vom caracteriza simplist un switch prin:

- Id - un număr natural
- Denumire - denumirea switch-ului (șir de caractere alocat dinamic)
- Adresa IPv4 - un număr pe 32 biți (4 octeți, separați de punct). Ex: 192.168.13.10
- Mod de funcționare - 2 moduri: mod stivă, mod individual

Pentru a putea fi administrate mai ușor, switch-urile se pot grupa în stive. Fiecare stivă are un singur switch principal. Administrarea switch-urilor se poate face individual sau prin intermediul switch-ului principal al stivei. Astfel configurarea unui switch se poate realiza prin

conectarea administratorului de rețea la switch-ul principal din stivă sau prin conectarea directă la switch-ul ce va fi configurat.

Switch-ul principal dintr-o stivă va fi mereu localizat la baza acesteia, iar restul switch-urilor vor fi aranjate după id. Astfel switch-ul cu id-ul cel mai mare se va afla mai aproape de baza stivei.

Switch-urile care nu lucrează în modulul stivă vor fi ținute într-o coadă sortate crescător după IPv4.

## 2. Cerință

Va trebui să executați un set de operații care se vor citi dintr-un fișier.

Operațiile pot fi:

**add** - adaugarea unui switch în rețea

**del** - scoaterea unui switch din rețea

**set** - setează mod de lucru pentru switch.

**ipmin** - afișarea celei mai mici adrese IPv4 a switch-urilor care funcționează în mod individual

**show** - afișarea cozii și a stivelor

## 3. Descrierea operațiilor și a datelor de intrare

Pe prima linie a fișierului de intrare se va afla numărul de stive din rețea.

Pe următoarele linii se găsesc operațiile care pot fi aplicate asupra switch-urilor.

Se va citi codul comenzii, iar în funcție de acesta se citește descrierea comenzii ca mai jos.

**add [id] [denumire] [IP] [funcționare] [id\_stiva] [principal]**

- Modul de **funcționare** poate avea valorile:
  - SINGLE -> funcționează individual
  - STACK -> funcționează în stivă
- Un switch poate fi principal sau secundar, în funcție de valorile argumentului **principal**:
  - NON\_BASE -> switch secundar
  - BASE -> switch principal

**Observații:**

1. Dacă switch-ul care va fi adăugat funcționează în mod stivă și este switch principal, atunci switch-ul care este principal înaintea inserării acestuia nu va mai fi principal.
2. Dacă un switch este inserat într-o stivă goală, atunci el va deveni switch principal.

**del id**

- Switch-ul va fi scos din rețea doar dacă acesta există

### **set [id] [functionare] [id\_stiva] [principal]**

1. Dacă `$functionare = modul curent de funcționare al switchului` atunci nu se efectuează nicio schimbare.
2. Dacă switch-ul principal este scos din stivă, locul lui este luat de switch-ul cu id-ul cel mai mare din acea stivă.
3. Dacă un switch este inserat în stivă ca switch principal atunci switch-ul care a fost principal până la acel moment își pierde acest rol, iar stiva trebuie refăcută (reordonată).

### **ipmin**

1. Dacă toate switchurile lucrează în mod stivă atunci se va afișa 0.

Ex: În loc de adresa 192.168.1.13 va fi afișat numărul 3232235789

### **show**

1. Se vor afișa pe o linie id-urile switch-urilor din coadă. Stivele se vor afișa în ordinea id-urilor. Switch-urile se vor afișa de la varful stivei la bază, fiecare pe câte o linie astfel: id IPv4 denumire.

## **4. Restricții și precizări:**

- O adresă IPv4 este de forma `xxx.xxx.xxx.xxx`,  $0 \leq xxx \leq 255$ . Ex: 192.168.1.13
- Un switch poate funcționa în modul stivă chiar dacă el este singurul switch din stivă
- Stivele vor fi indexate de la 0 la `numar_stive-1`
- Denumirile switchurilor sunt șiruri cu maxim 255 de caractere și nu conțin spații
- Datele de intrare nu trebuie validate
- Programul va fi rulat astfel: **`./tema2 in_file out_file`**
- Comenzile se citesc din fișierul **`in_file`**, iar rezultatele se scriu în fișierul **`out_file`**
- Se garantează că switchurile vor avea id-uri și IP-uri unice.
- Dacă un switch principal dintr-o stivă alege să treacă în modul de funcționare individual, atunci funcția de switch principal în acea stivă va fi preluată de switch-ul cu id-ul cel mai mare din acea stivă, iar stiva va fi restabilită.
- Stivele și cozile vor fi implementate ca **liste generice simplu înlantuite**.
- **Nu** aveți voie să iterați prin conținutul stivelor sau al cozii nici la afișarea conținutului acestora.
- **Nu** aveți voie să rețineți switch-urile în alte structuri de date diferite de stive și cozi.
- **Nu** aveți voie cu variabile globale
- Makefile-ul va conține regulile **`build`** și **`clean`**.

## 5. Exemplu

1 add 3 possum 192.168.81.54 SINGLE add 1 ranger4900 192.168.78.215 STACK 0 BASE add 4 possum 192.168.40.80 STACK 0 NON_BASE show add 0 ranger4900 192.168.82.151 STACK 0 BASE add 2 ranger4900 192.168.54.106 STACK 0 BASE show set 4 SINGLE set 1 STACK 0 BASE show del 1 del 0 show ipmin	{3 } 0: 4 192.168.40.80 possum 1 192.168.78.215 ranger4900  {3 } 0: 0 192.168.82.151 ranger4900 1 192.168.78.215 ranger4900 4 192.168.40.80 possum 2 192.168.54.106 ranger4900  {4 3 } 0: 0 192.168.82.151 ranger4900 2 192.168.54.106 ranger4900 1 192.168.78.215 ranger4900  {4 3 } 0: 2 192.168.54.106 ranger4900  ipmin=3232245840
--	--

## 6. Notare

- **85 de puncte** obținute pe testele de pe vmchecker
- **10 puncte:** coding style (a se vedea [1]), lipsa warningurilor și a liniilor mai lungi de 80 de caractere, folosirea incorectă de pointeri, neverificarea codurilor de eroare, utilizarea unor metode ce consuma resurse in mod inutil (alocare de memorie), neeliberarea resurselor folosite (eliberare memoriei alocate, stergerea fisierelor temporare, inchiderea fisierelor), alte situatii nespecificate aici, dar considerate inadecvate
- **5 puncte:** README - va contine detaliile de implementare a temei, precum si punctajul obtinut la teste (la rulara pe calculatorul propriu)
- **Bonus: 20 de puncte** pentru soluțiile care nu au memory leak-uri (bonusul se acorda daca testul a trecut cu success)

- Temele care nu compileaza, nu ruleaza sau obtin punctaj 0 la teste, indiferent de motive, vor primi punctaj 0
- NU copiați!!!

## 7. Reguli de trimitere a temelor

- A se vedea și Regulile generale de trimitere și punctare a temelor [2].
- Temele vor trebui incarcate atat pe vmchecker (in secțiunea Structuri de Date seria CB: **SD-CB**) cat si pe cs.curs.pub.ro, in sectiunea aferenta Temei 1.
- Arhiva cu rezolvarea temei trebuie să fie .zip și să conțină:
  - fisiere surse (fiecare fisier sursa creat sau modificat va trebui sa inceapa cu un comentariu de forma:  
**/\* NUME Prenume - grupa \*/**
  - fisier README care sa contina detalii despre implementarea temei
  - fișier Makefile cu doua reguli: Fișierul pentru make trebuie denumit obligatoriu Makefile și trebuie sa contina urmatoarele reguli:
    - build, care va compila sursele si va obtine executabilul, cu numele tema1.
    - clean, care va sterge executabilele generate.
  - arhiva nu trebuie să contina decat fisierele sursa (nu se accepta fișiere executabile sau obiect)
  - daca arhiva nu respecta specificațiile de mai sus nu va fi acceptata la upload și tema nu va fi luata in considerare