Tema 2 Cryptography

Termen de predare: **sâmbătă, 9 aprilie 2011, ora 23:59**

Last modified: 03-04-2011
~~Last modified: 02-04-2011~~
~~Last modified: 28-03-2011~~

Scopul temei

Aprofundarea următoarelor noţiuni:

- semafoare
- cozi de mesaje
- memorie partajată

Enunţ

Să se implementeze un protocol gateway - server folosind mecanisme IPC. Serverele deţin o metoda de decriptare, în care se fac decriptari comandate de mesajele primite de la gateway. Gateway-ul primeste comenzi de decriptare de la mai multi clienti. Arhitectura contine 4 servere de decriptare si un gateway.

Gateway-ul este comandat prin mesaje primite pe coada de mesaje denumita "gateway" pe Linux, respectiv MailSlot denumit "\\.\mailslot\gateway" în Windows, iar acesta trimite mesajul catre serverele specifice pentru a fi decriptat. Un mesaj este o structura de tipul:

```
typedef struct command {
  char name[8]; /* Numele memoriei partajate si a semaforului. */
  int dim; /* Dimensiunea memoriei partajate. */
  short crypt[17]; /* Contine ordinea serverelor care vor face decriptarea. Secventa de servere se termina cu -1*/
} Command,*PCommand;
```

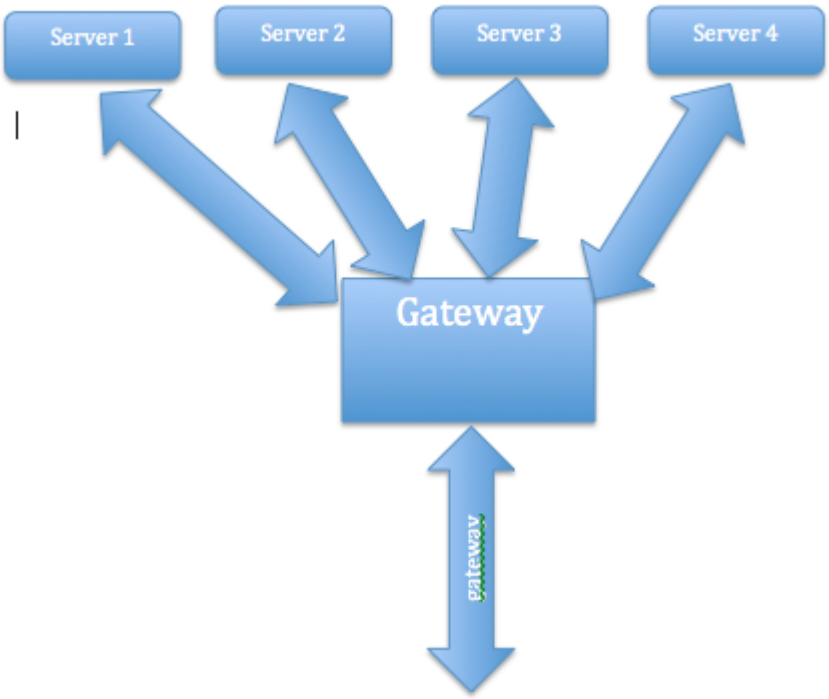
Operatiile trebuie sa fie blocante (clientul nu trebuie eliberat pana nu a fost decriptat mesajul). Pentru aceasta clientul care trimite cererea gateway-ului deschide semaforul cu numele trimis in structura cu valoarea initiala 0, si face un apel de tipul acquire pe acesta. Gateway-ul va elibera acest client dupa ce mesajul decriptat se afla in memoria partajata.

Serverele vor folosi biblioteca partajata din enunt si se vor porni sub forma:

```
  ./server N
```

N - este un numar din multimea {1, 2, 3, 4} :

- 1 - reprezinta un server care decripteaza base64, va folosi din biblioteca partajata decrypt_base64;
- 2 - reprezinta un server care decripteaza des, va folosi din biblioteca partajata decrypt_des;
- 3 - reprezinta un server care decripteaza bf, va folosi din biblioteca partajata decrypt_bf;
- 4 - reprezinta un server care decripteaza rc2, va folosi din biblioteca partajata decrypt_rc2;



Formatul de intrare / de afişare

Mesajele se afla in zona de memorie partajata astfel:

- primii 4 octeti sunt un numar (int) care reprezinta dimensiunea mesajului criptat/ decriptat.
- urmatorii N octeti sunt mesajul criptat/ decriptat.

Atentie se garanteaza ca lungimea acestui mesaj nu va depasi dimensiunea zonei partajate.

Precizări generale

- Clientul trimite o lista de decriptari ce trebuie aplicate asupra mesajului. Este important ca decriptarile sa fie aplicate in ordinea prevazuta de client.
- Mesajul initial (inainte de criptare) are o lungime mai mica de 20 de caractere.
- Se garanteaza ca mesajele criptate ca au o lungime mai mica de 1000 de bytes.
- Pot exista mai multi clienti activi simultan. (Un gateway poate primi concomitent mesaje de la mai multi clienti)
- Pentru sincronizare se vor folosi numai semafoare, atât pe Linux, cât şi pe Windows.
- Comunicarea intre gateway si servere se face numai prin cozi de mesaje (MailSlots în Windows). Mesajul propriu-zis (cel ce va fi decriptat) trebuie pastrat in memoria partajata
- Trebuie implementat doar Gateway-ul si Serverul (cu optiunea de a putea fi pornit pentru toate cele 4 cazuri).
- In cazul in care se primeste o structura cu campul **name** == "exit" atunci gateway-ul trebuie sa comande oprirea serverelor si sa se opreasca si el.
- Nu uitati sa faceti unlink la semafoare, zone de memorie partajata, cozi de mesaje in cazul in care programul esueaza. Este recomandat sa faceti unlink inainte de crearea lor.
- Este recomandata folosirea masinii virtuale pentru realizarea acestei teme. In cazul in care nu sunteti familiarizat cu vim, puteti monta respectiva partitie pe Linux (Places/Connect to server.../SSH) sau edita fisierul prin winscp (Windows).
- Comanda pentru exit: ./client "exit" "exit" -1

Precizări Windows

- Tema se va rezolva folosind doar funcţii Win32. Se pot folosi de asemenea şi funcţiile de formatare `printf`, `scanf`, funcţiile de alocare de memorie `malloc`, `free` şi funcţiile de manipulare a şirurilor de caractere (`strcat`, `strdup` etc.)
- Deoarece semafoarele si memoria partajata nu pot avea acelasi nume pe windows, acestea vor fi :
  - 'name'_sem - semaforul

- 'name'_shm - memoria partajata. Unde name este cel din structura primita pe coada de mesaje.

Precizări Linux

- Tema se va rezolva folosind doar funcţii POSIX. Se pot folosi de asemenea şi funcţiile de formatare printf, scanf, funcţiile de alocare de memorie malloc, free şi funcţiile de manipulare a şirurilor de caractere (strcat, strdup, etc.)
- Pentru partea de IPC se vor folosi funcţiile POSIX (shm_*, mq_*, sem_*) şi nu cele SysV!

Testare

Pentru simplificarea procesului de corectare al temelor, dar şi pentru a reduce greşelile temelor trimise, corectarea temelor se va face automat cu ajutorul unor teste publice (Linux, Windows).

În urma compilării temei trebuie să rezulte două executabile:

- **server** (respectiv server.exe pentru Windows)
- **gateway** (respectiv gateway.exe pentru Windows).

Numele acestor executabile trebuie sa fie respectat.

Atenţie! Terminarea incorecta a programului server poate lăsa fişierele corespunzătoare semafoarelor deschise. Verificaţi în /dev/shm dacă întâmpinaţi probleme la creearea semafoarelor.

Nota maximă obţinută din teste este 9. Se acordă 1 punct din oficiu.

Nota mai poate fi modificată prin depunctări suplimentare:

- Lista generala de depunctari
- `-0.1` diverse alte probleme constatate în implementare

Materiale ajutătoare

Cursuri:

- Curs 5

Laboratoare:

- Laborator 1
- Laborator 5

Arhiva cu bibliotecile partajate (compilate pe masiniile virtuale)

- shared library

Teste

- linux
- windows

Pagina de Upload:

- upload

FAQ

- **Q:** Temele se pot face în C++?
  - **A:** Da.

- **Q:** Restrictiile din enunt trebuie verificate?
  - **A:** Nu.

Lista de discuţii

Pentru întrebări sau nelămuriri legate de temă puteţi căuta, consulta sau trimite un mail pe lista de discuţii (trebuie să fiţi înregistraţi).