

## Tema 1

În această temă, se consideră un sistem client-server în care mai mulți clienți vor să acceseze resursele unui server, iar acesta trebuie să implementeze o politică de control al accesului. Atât serverul cât și clientul vor trebui implementați.

Resursele vor fi organizate în mod ierarhic, ca într-un sistem de fișiere. De exemplu: **/bob/teme/psd/a.java** va fi resursa **a.java** în sistemul ierarhic **/bob/teme/psd**. O resursa poate fi de două tipuri: director și fișier obișnuit. Un director poate conține mai multe fișiere obișnuite. Fiecare fișier va avea un nume și o valoare de tip string și va avea asociate două permisiile: **read** și **write**.

Fiecare utilizator are un nume și o parolă. Se presupune că utilizatorul **bob** are asociat directorul **/bob** în care are toate drepturile (**read** și **write**). **Bob** poate schimba drepturile de acces ale resurselor pe care le deține pentru ceilalți utilizatori. Default, un utilizator nu poate accesa resursele create de alt utilizator, fără ca acesta să îi dea voie în mod explicit. Un fișier va moșteni drepturile de la directorul părinte, dacă nu le are pe ale sale proprii. Fiecare director poate avea drepturi diferite.

Serverul va trebui să implementeze această politică de acces în condițiile în care mai mulți clienți pot trimite cereri de acces. Serverul trebuie să poată suporta mai mulți clienți în paralel, deci trebuie să fie multi-threaded. Clientul poate trimite următoarele tipuri de cereri. Toate aceste cereri trebuie verificate de către server dacă pot sau nu fi executate conform politicii de securitate.

1. **createResursa**(String user, String parola, String numeResursa, int tip, String valoare)

- a. Dacă **numeResursa** există deja, serverul trebuie să returneze eroare.
- b. Tip poate fi 0 (director) sau 1 (fișier). Dacă este fișier, atunci valoarea va fi asignată acestei resurse.
- c. Numai utilizatorul care este owner în acel director (de exemplu **Bob** pentru orice din **/bob**) are voie să facă asta.

2. **readResursa**(String user, String parola, String numeResursa)

- a. Dacă nu există resursa, trebuie să returneze eroare.
- b. Politică de securitate trebuie analizată și să se returneze eroare dacă cererea nu este autorizată.
- c. Dacă este director, trebuie să returneze ce se găsește în acel director

3. **writeResursa**(String user, String parola, String numeResursa, String valoare)

- a. Dacă nu există resursa, trebuie să returneze eroare.
- b. Politică de securitate trebuie analizată și să se returneze eroare dacă cererea nu este autorizată.

4. **changeRights**(String user, String parola, String numeResursa, String rights)

- a. Dacă nu există resursa, trebuie să returneze eroare.

b. Politica de securitate trebuie analizata si sa se returneze eroare daca cererea nu este autorizata.

**Punctaj:**

Compilare program, rulare: 4p

Verificare server suporta mai multi clienti in paralel: 2p

Rulare TestCases: 4p

**Termen de predare: 1 Noiembrie 2016.**

**Testcases**

Aceste comenzi trebuie rulate in aceasta ordine. Presupun un sistem cu doi utilizatori, Alice si Bob, initial doar cu directoarele implicite: /bob si /alice. Parola pt Alice o presupun alicia si a lui Bob bob:

1. creareResursa(alice, alice, /bob/cursuri, 0) **returneaza NotAuthorized**
2. creareResursa(alice, alice, /alice/cursuri, 0) **returneaza ok**
3. creareResursa(alice, alice, /alice/cursuri, 0) **returneaza AlreadyExisting**
4. creareResursa(alice, alice, /alice/cursuri, 1) **returneaza AlreadyExisting**
5. readResursa(bob, bob, /alice/cursuri) **returneaza NotAuthorized**
6. readResursa(alice, alice, /alice/cursuri) **returneaza ok, ""**
7. changeRights(bob, bob, /alice/cursuri, rw) **returneaza NotAuthorized**
8. changeRights(alice, alice, /alice/cursuri, rw) **returneaza ok**
9. readResursa(bob, bob, /alice/cursuri) **returneaza ok, ""**
10. creareResursa(alice, alice, /alice/cursuri/a.java, 1, "Test") **returneaza ok**
11. readResursa(bob, bob, /alice/cursuri) **returneaza ok, "a.java"**
12. readResursa(bob, bob, /alice/cursuri/a.java) **returneaza ok, "Test"**
13. changeRights(alice, alice, /alice/cursuri/b.java, rw) **returneaza NotExisting**
14. changeRights(alice, alice, /alice/cursuri/a.java, "") **returneaza Ok**
15. readResursa(bob, bob, /alice/cursuri/a.java) **returneaza NotAuthorized**

**Important!**

**Pentru implementare va puteti alege limbajul de programare dorit.**