Documentație Proiect CopyService

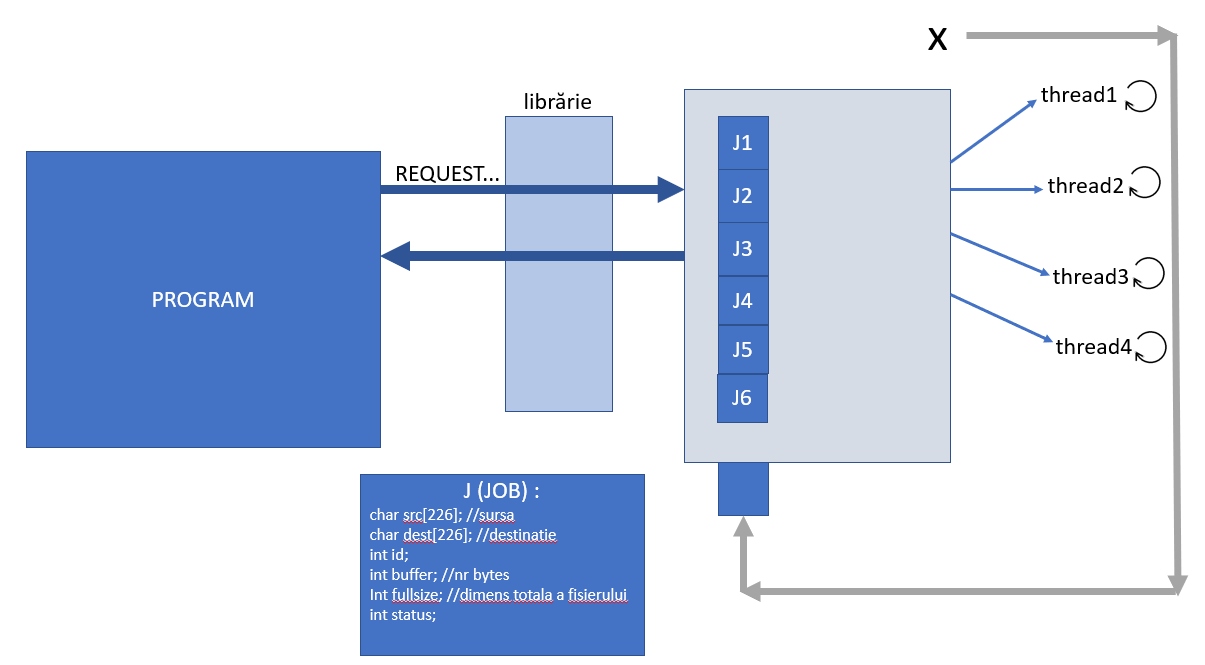
Proiectul descrie un Copy Service realizat printr-un daemon în userspace, ce permite copierea asincronă de fișiere.

În scopul realizării lui, am inclus o librărie ce cuprinde următoarele API-uri:

* Crearea unui job de copiere
* **struct job CreateJob( const char\* src, const char\* dest);**
* Anularea unui job de copiere
* **void CancelJob();**
* Întreruperea unui job de copiere
* **void PauseJob();**
* Obținerea de informație
* **void StartJob();**
* Enumerarea joburilor de copiere
* **void PrintJob();**

Daemonul în sine este serverul, care așteaptă comenzi (joburi) de la mai mulți clienți. Acești “clienți” sunt, de fapt, programe de test ce trimit comenzi daemonului folosing librăria pusă la dispoziție de noi.

Dinamica proiectului s-a bazat pe următoarea schemă:



Pentru a reuși să comunicăm cu daemonul, ne folosim de **unix sockets**, astfel că acesta așteaptă comenzile de la “clienții” ce se conectează în paralel la socket.

Pentru executarea joburilor am folosit conceptul de **thread pool,** însemnând că avem un număr limitat de threaduri (prestabilit în fișierul config pentru daemon) care stau într-o buclă infinită și așteaptă joburi noi, care vor fi adăugate într-o coadă.

La partea de sincronizare, threadurile vor fi controlate de un mutex.

Împărțirea proiectului

* *Directorul include* – am inclus toate headerele: job.h (enumerarea tuturor tipurilor de joburi si struct job), queue.h (struct queue si antentul functiilor – push, front, pop, empty), spellcaster.h (contine antetul funcțiilor folosite de job)
* *Directorul src*
* daemon.c – codul daemonului
* queue.c – corpul functiilor folosite de coadă
* spellcaster.c – corpul funcțiilor folosite de job
* *Directorul tests* – codul programelor ce reprezinta “clienții”

Proiectul conține și un Makefile care permite rularea tuturor programelor (comanda make all).