# Notite Thread-uri Reading

* Thread-urile sunt fire de executie din cadrul aceluiasi process;
* Thread-urile mostenesc de la procesul parinte contextul si zonele de date, dar elemental exclusive este zona de stiva, care este specifica fiecarui thread.
* Thread-urile pot fi executate concurentiale si in paralel.
* Pentru a folosi thread-urile POSIX, se foloseste libraria <pthread.h>.
* Pentru a crea thread-uri, se foloseste functia pthread\_create(pthread\_t \*id, pthread\_attr, void \*(\*func)(void\*), void \*arg);
  + Id: reprezinta descriptorul thread-ului.
  + Attr: reprezinta atributele pe care le va avea thread-ul. In general se vor dori atributele implicite, astfel incat pointer-ul va fi NULL;
  + Func: reprezinta o referinta la functia executa de thread. Functia va returna void si va primi un pointer catre o zona de memorie unde se afla datele.
  + Arg: pointer catre argumente functiei func;
* Pentru terminarea unui thread se foloseste functia pthread\_exit(int \*status);
  + Daca pointer-ul status nu este NULL sau 0, se pot intoarce rezultate catre procesul parinte al thread-ului. Rezultatele pot transmite anumite erori sau detalii legate de executie.
* Desi thread-urile admit executia paralela si concurenta, de cele mai multe ori executia acestor thread-uri trebuie sincronizata.
* Cea mai simpla metoda de sincronizare este pthread\_join(pthread\_t tid, void \*\*status);
  + Suspenda executia thread-ului parinte pana la finalizarea thread-ului tid;
  + Statusul primeste valoare de la pthread\_exit din interiorul thread-ului.
* Mecanismele de sincronizare au ca scop controlarea accesului thread-urilor asupra resurselor critice(resursele ce pot fi modificate de mai multe thread-uri).
* Exemple de mecanisme de sincronizare sunt: mutex si variabile conditionale;
* Mutex-ul permite blocarea controlata la accesul resurselor critice de catre un thread. Un thread trebuie sa blocheze un mutex pentru a accesa datele. In cazul in care este deja blocat, asteapta. La finalizarea instructiuniilor, trebuie deblocat acest mutex. Blocarea si deblocarea incorecta poate conduce la dead-lock.
* Variabila conditionala este defapt un mutex caruia operatiile de blocare si deblocare sunt legat de o anumita conditie impusa de utilizator.