*My\_Barrier Implementation Assigment*

1. Pentru **implementarea tipului ‘my\_barrier\_t’** am folosit un struct ce contine o variabila coditionala, aceasta fiind necesara pentru ca primele thread-urile sa astepte si sa isi mentina starea pena la primirea flag-ului de la ultimul ajuns, si un mutex pentru a preveni problemele de tipul ‘race conditions’ unde 2 threads pot accesa simultan datele barierei, anume un ‘contor’ ce ne spune la momentul curent cate thread-uri au ajuns la bariera, si un ‘total’ ce reprezinta numarul total de thread-uri ce sunt asteptate.
2. Pentru **implementarea celor 3 functii necesare,anume my\_barrier\_init/wait/destroy** am folosit diverse abordari:
   1. *my\_barrier\_init* seteaza contorul current la 0, iar pe cel total la nr primit ca argument. De asemenea, un mutex si o variabila conditionala sunt initializate pentru viitoarele operatii necesare unei bariere.
   2. *my\_barrier\_wait* are cam prim pas un mutex\_lock pentru a evita posibile conflicte intre thread-uri. Folosim la fiecare apelare de metoda POSIX o functie de verificare aditionala, pentru a fi siguri ca gestionam in mod correct fluxul de date. Urmeaza sa incrementam count pentru fiecare thread ajuns, dupa care o verificare daca toate thread-urile au ajuns, resetam count, afisam un mesaj pe consola, cu rolul de a ajuta la lizibilitatea textului din consola, separand task-urile intre ele, iar apui variabila conditionala permite thread-urilor sa porneasca executia din starea in care au ajuns la bariera.
   3. *my\_barrier\_destroy* are ca unic scop doar eliberarea memoriei ocupate de mutex si variabila conditionala, prin apelarea functiilor destroy din POSIX.
3. Pentru***implementarea scenariului de folosire(*daily life of Jones Family)** am folosit o bucla pentru a simula un numar da de iteratii dorite, la anumite puncta din program, cum ar fi micul dejun, unde toate thread-urile asteapta ca fiecare sa isi inceapa executia si sa execute taskul simultan. In plus, pe langa bariere ce asteapta toate thread-urile, avem si unele mai specific, deoarece asteapta numai un numar restrains de thread-uri, si chiar unele specific, cum ar fi cazul Alice-Bob pentru a studia matematica impreuna.