

TP1 AP3 : file d'attente

Une file informatique permet de modéliser simplement une file d'attente, de type bureau de poste, comme suit :

- Un unique guichet, ouvert 8h00 consécutives ($8 \times 60 \times 60$ sec.)
- Les clients arrivent et font la queue.
- La probabilité qu'un client arrive durant l'intervalle $[t..t + 1[$ est p (passé en argument lors de l'appel du programme)
- Le temps pris pour servir un client est dans l'intervalle $[30...300[$.
- Les clients partent exaspérés au bout d'un temps aléatoire compris dans $[120..1800[$

Objectif du TP : simuler le fonctionnement de ce bureau de poste pour estimer le rapport $r = \text{client partis exaspérés} / \text{clients venus}$, en fonction de la probabilité p . Pour cela, voici les différentes étapes à réaliser :

- 1) Proposer une structure pour représenter un client
- 2) Ecrire le fichier client.c (et son en-tête .h) comprenant **au moins** la structure, la fonction de création d'un client et la fonction uniform (son code est donné ci-dessous) qui permet de calculer le temps au bout duquel le client partira exaspéré

```
int uniform(int min , int max ) {  
    int d = max-min ;  
    return ( (int) ( ( ( double ) rand ( ) / ( double )RAND_MAX) * d)+min ) ; }
```

- 3) Proposez une structure files pour représenter la file d'attente
- 4) Écrire le fichier fifo.c (et son en-tête .h) avec les fonctions suivantes :
 1. files createFifo() //création d'une file
 2. add_Fifo(files f,client c) //ajout d'un client dans la file
 3. client removeFifo(files f) //retrait d'un client de la file
 4. bool isemptyFifo (files f) //test de la file vide
 5. bool isfull(files f) //test de la file pleine
- 5) Écrire le fichier simulation.c (et son en-tête .h), pour cela déterminer/écrire :
 1. les constantes du problème
 2. **l'algorithme du fonctionnement du bureau de poste**
 3. la fonction (donnée ci-dessous) qui calcule la survenue ou pas (un booléen) d'un événement (l'arrivée d'un client) de probabilité p dans l'intervalle $[0-1]$

```
bool occur(double p) { return( ( ( double ) rand( ) ) /RAND_MAX)< p ; }
```

4. on pourra réutiliser la fonction occur vu au 2°) afin de calculer la durée (aléatoire) pour servir un client
5. Enfin écrire la fonction de simulation : double simul(double p) qui retourne le rapport r
- 6) Écrire le programme principal (dans un autre fichier) d'appel de la simulation (lors de l'appel de ce programme principal on passe en paramètre la probabilité p , sinon on choisira une valeur p de votre choix)

N'oubliez pas de faire valider vos algorithmes ou structures avant de passer au codage !!!
Bon courage.