

Programmation OpenMP

1 – On a un tableau d'échantillons sur 10bits, c-à-d dont la valeur est comprise entre 0 et 1023 ($2^{10} = 1024$).

On veut en réaliser l'**histogramme**, c-à-d compter le nombre de fois où chaque valeur apparaît :

- ▷ on parcourt les cases du tableau de valeurs ;
- ▷ pour chaque valeur rencontrée on augmente le nombre d'occurrences associé.

tableau échantillons	3	9	0	5	4	4	5	0	6	0	4	4
occurrences	3	0	0	1	4	2	1	0	0	1		
valeur	0	1	2	3	4	5	6	7	8	9		

} *histogramme*

Soit le programme Python créant les données :

```
#!/usr/bin/python3

import random, struct, sys

nom_fichier = 'data.bin'
nombre_données = 1024
taille_échantillon = 2**10

try:
    f = open(nom_fichier, 'wb')
except Exception as e:
    print(e.args)
    sys.exit(1)

for i in range(0,nombre_données):
    f.write(struct.pack('i', random.randint(0,taille_échantillon-1)))
f.close()
```

Celui calculant l'histogramme :

```
#!/usr/bin/python3

import random, struct, sys

nom_fichier = 'data.bin'
nombre_données = 1024
taille_échantillon = 2**10

try:
    f = open(nom_fichier, 'rb')
except Exception as e:
    print(e.args)
    sys.exit(1)

tableau_données = [0] * nombre_données
rang = 0
while 1:
    entier = f.read(4)
    if not entier:
        break
    valeur = struct.unpack('i', entier)[0]
    tableau_données[rang] = valeur
    rang += 1
f.close()

histogramme = [0] * taille_échantillon
for v in tableau_données:
    histogramme[v] += 1
print(histogramme[:30])
```

Écrire un programme OpenMP **optimisé** réalisant ce travail.