

Au lieu d'appeler gcc plusieurs fois, un Makefile permettra de construire les fichiers sources et de produire l'exécutable :

```
CC = gcc
TARGET = hist
OBJECTS = $(patsubst %.c,%.o, $(wildcard *.c))

$(TARGET): $(OBJECTS)
    @echo "*** Liaison de l'Executable"
    $(CC) $(OBJECTS) -o $(TARGET)

clean:
    @rm -f *.o *~

veryclean: clean
    @rm -f $(TARGET)

%.o: %.c
    @echo "*** Compilation du source C"
    $(CC) -c $<
```

Plus d'informations sur make se trouvent dans le n°7. Placez les deux fichiers .c dans le même répertoire, avec histogram.h et Makefile. Tapez ensuite make pour construire l'exécutable. Le programme lancé va produire un fichier contenant les sommes des poids dans les classes, en-dessous et au-dessus. Le format est choisi pour permettre de tracer l'histogramme avec gnuplot :

```
gnuplot
plot 'Hist.txt' with boxes
```

Comme dans le précédent tutoriel, la commande pour tracer peut être intégrée au programme. Pour enregistrer le graphique en png, tapez :

```
set term png
set output 'Hist.png'
replot
```

après la commande de traçage d'origine.

Le défi

Utilisez cet article et le précédent pour faire l'histogramme de la charge système sur 30 minutes. Ajoutez des fonctions à histogram.h et histogram.c pour calculer la moyenne et l'écart-type de la distribution. La moyenne d'un histogramme se calcule à partir de :

```
xMean = 0.;
for (i=1;i<=nBins;i++) { /* Ignore débordements */
    xMean += binContents[i]/nBins;
}
```

L'écart-type est donné par :

```
xStdDev = 0.;
for(i=1;i<=nBins;i++) { /* Ignore débordements */
    xStdDev += pow(xMean - binContents[i],2)/(nBins-1);
}
if(xStdDev > 0.) xStdDev = sqrt(xStdDev);
```

La solution du problème sera donnée la prochaine fois.

Article de W. H. Bell