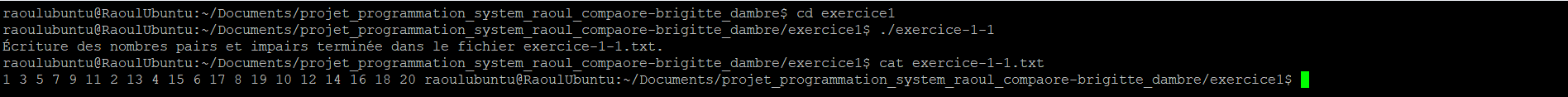
**RAPPORT DU PROJET DE PROGRAMMATION SYSTEME EXERCICE 1**

**1)**

Ecrire un programme qui ouvre un fichier à l’aide de la primitive open. Par la suite le programme crée deux processus légers. Le premier écrira des nombres pairs et le second des nombres impairs

Notre programme crée deux threads distincts, l'un chargé d'écrire dans un fichier tous les nombres pairs jusqu'à une valeur maximale donnée, et l'autre chargé d'écrire tous les nombres impairs jusqu'à cette même valeur maximale. Il utilise la bibliothèque POSIX Threads pour la gestion des threads. Les threads partagent un descripteur de fichier et une valeur maximale à écrire, transmis via une structure ThreadData. Le fichier de sortie est ouvert en mode écriture, et une fois les threads terminés, le descripteur de fichier est refermé. Le programme fournit un exemple pédagogique de l'utilisation des threads pour effectuer des tâches concurrentes en C.

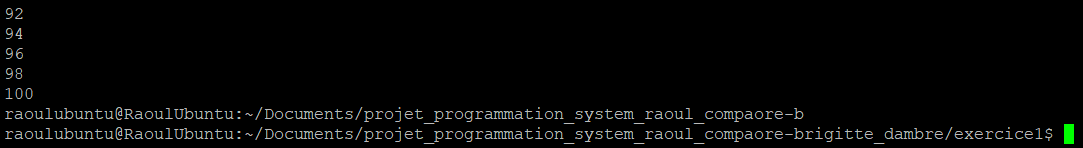
Captures d’ecrans :

**2)**

Ecrire un programme qui ouvre un fichier à l’aide de la primitive open. Par la suite le programme crée deux processus légers. Le premier écrira des nombres pairs et le second des nombres impairs. On synchronisera les deux processus légers pour que les nombres soient écrits dans l’ordre croissant.

Le programme crée deux threads qui écrivent simultanément des nombres pairs et impairs dans un fichier ("exercice-1-2.txt"). Il utilise un verrou mutex pour garantir un accès sûr aux données partagées entre les threads, assurant ainsi une écriture cohérente dans le fichier.

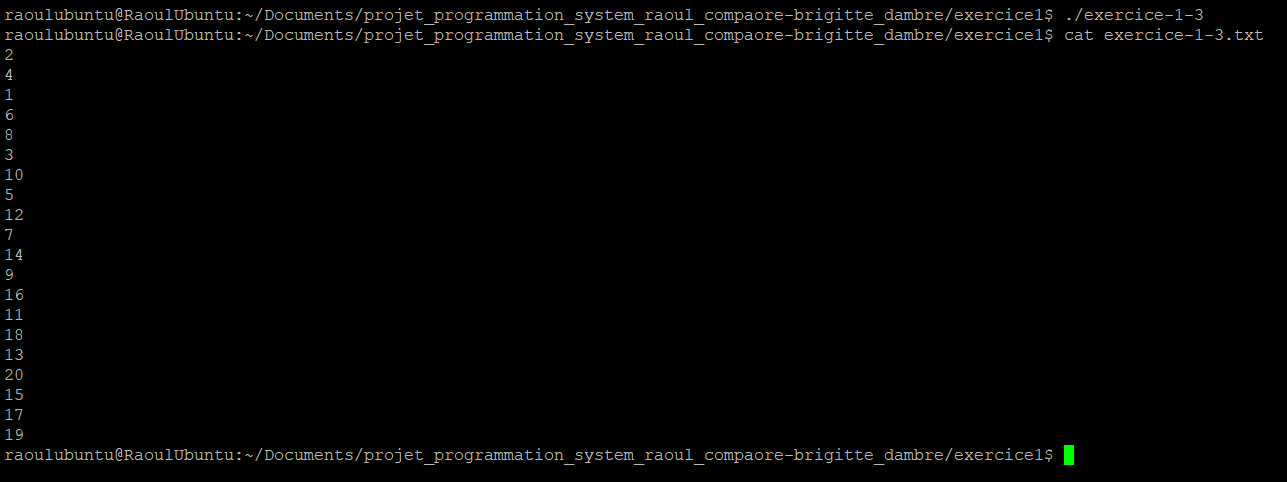
Captures d’ecrans :



**3)**

Ecrire un programme qui ouvre un fichier à l’aide de la primitive open. Par la suite le processus (père) crée un autre processus (fils). Le premier (le père) écrira des nombres pairs et le second (le fils) des nombres impairs.

Le programme crée un fichier "exercice-1-3.txt" et utilise un processus fils pour écrire simultanément des nombres pairs et impairs dans ce fichier. Le père écrit les nombres pairs de 2 à 20, et le fils écrit les nombres impairs de 1 à 19.

Captures d’ecrans :  


**4)**

Ecrire un programme qui ouvre un fichier à l’aide de la primitive open. Par la suite le processus (père) crée un autre processus (fils). Le premier (le père) écrira des nombres pairs et le second (le fils) des nombres impairs. On synchronisera les deux processus pour que les nombres soient écrits dans l’ordre croissant

Le programme crée un fichier "exercice-1-4.txt" et utilise un processus fils pour écrire simultanément des nombres pairs et impairs dans ce fichier. Il utilise également un sémaphore pour synchroniser l'accès concurrent au fichier, assurant une écriture cohérente. Le père écrit les nombres pairs de 2 à 20, et le fils écrit les nombres impairs de 1 à 19.

