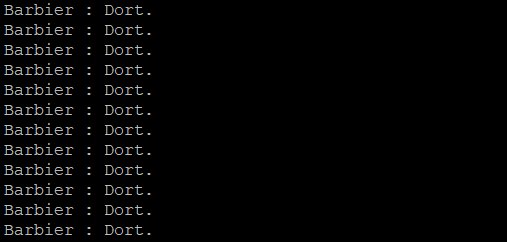
RAPPORT DU PROJET DE PROGRAMMATION SYSTEME EXERCICE 2

**1)**  
Ecrivez le code du client et du barbier sans vous préoccuper de synchronisation, mais simplement des opérations que veulent réaliser les processus. En particulier, ignorez pour l’instant que le barbier dort parfois.

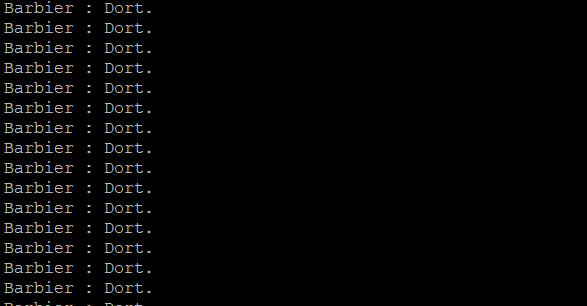
Le programme crée un barbier et deux clients en utilisant des threads. Le barbier et les clients partagent l'accès à une salle d'attente limitée (**MAX\_CHAIRS**). Les clients entrent dans la salle d'attente si elle n'est pas pleine. Si le barbier dort, un client le réveille. Le barbier invite un client dans le salon, le rase, puis retourne dormir si la salle d'attente est vide. Le programme se termine après un court délai d'exécution, illustrant le processus de coiffure et la gestion des clients par le barbier.



**2)**

Il s’agit maintenant d’ajouter les synchronisations nécessaires au programme écrit à la question précédente. Le premier problème `a résoudre est une condition de compétition entre les clients lorsqu’ils rentrent dans la salle d’attente. Corrigez ce problème.

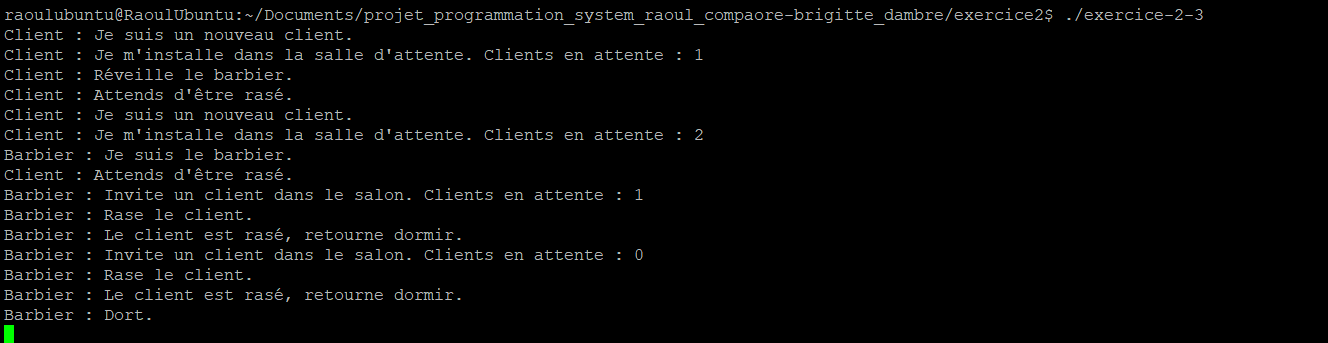
Le programme crée un barbier et deux clients en utilisant des threads. Les clients entrent dans la salle d'attente si elle n'est pas pleine. Si le barbier dort, un client le réveille. Le barbier invite un client dans le salon, le rase, puis retourne dormir si la salle d'attente est vide. Le programme se termine après un court délai, illustrant le processus de coiffure et la gestion des clients par le barbier. Les informations sur le nombre de clients en attente sont également affichées.



**3)**

Assurez-vous ensuite que le barbier ne commence pas à couper les cheveux tant que le client n’est pas prêt.

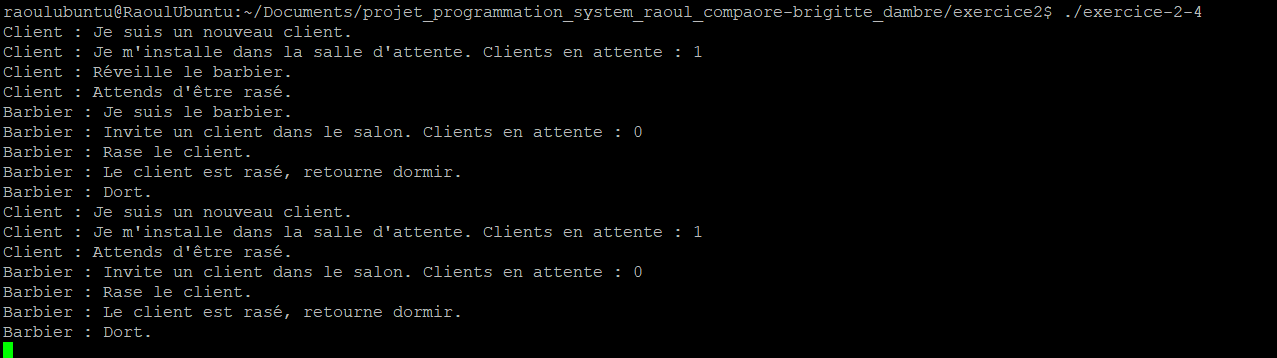
Le programme crée un barbier et deux clients en utilisant des threads. Les clients entrent dans la salle d'attente si elle n'est pas pleine. Si le barbier dort, un client le réveille en signalant une condition. Le barbier invite un client dans le salon, le rase, puis retourne dormir si la salle d'attente est vide. Le programme se termine après un court délai, illustrant le processus de coiffure et la gestion des clients par le barbier. Les informations sur le nombre de clients en attente sont également affichées.



**4)**

Enfin, assurez-vous ensuite que le client ne s’assoit pas sur le siège tant que le barbier n’est pas prêt.

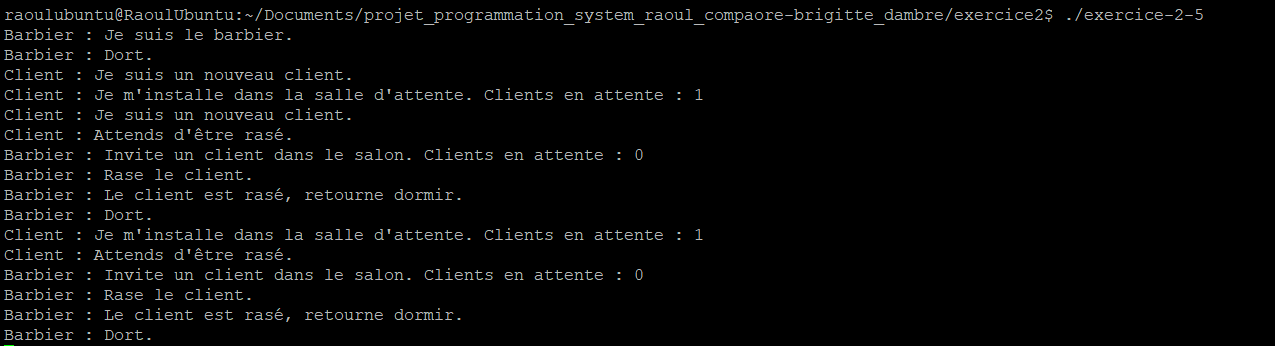
Le programme crée un barbier et deux clients en utilisant des threads. Les clients entrent dans la salle d'attente si elle n'est pas pleine. Si le barbier dort, un client le réveille en signalant une condition. Le barbier invite un client dans le salon, le rase, puis retourne dormir si la salle d'attente est vide. Le programme se termine après un court délai, illustrant le processus de coiffure et la gestion des clients par le barbier. Les informations sur le nombre de clients en attente sont également affichées. La communication entre le barbier et le client se fait à l'aide de deux conditions distinctes (**clientReady** et **barberReady**) pour gérer les différentes étapes du processus.



**5)**

Proposer une simulation (programme C) de votre solution en utilisant des threads (processus légers).

Le programme crée un barbier et deux clients en utilisant des threads. Les clients entrent dans la salle d'attente si elle n'est pas pleine. Si le barbier dort, un client le réveille en signalant une condition. Le barbier invite un client dans le salon, le rase, puis retourne dormir si la salle d'attente est vide. Le programme se termine après un court délai, illustrant le processus de coiffure et la gestion des clients par le barbier. Les informations sur le nombre de clients en attente sont également affichées. La communication entre le barbier et le client se fait à l'aide de deux conditions distinctes (**clientReady** et **barberReady**) pour gérer les différentes étapes du processus.



**6)**

Proposer une simulation (programme C) de votre solution en utilisant des processus lourds

Le programme crée un barbier et deux clients en utilisant des threads. Les clients entrent dans la salle d'attente si elle n'est pas pleine. Si le barbier dort, un client le réveille en signalant une condition. Le barbier invite un client dans le salon, le rase, puis retourne dormir si la salle d'attente est vide. Le programme se termine après un court délai, illustrant le processus de coiffure et la gestion des clients par le barbier. Les informations sur le nombre de clients en attente sont également affichées. La communication entre le barbier et le client se fait à l'aide de deux variables conditionnelles (**clientCond** et **barberCond**) pour gérer les différentes étapes du processus.

