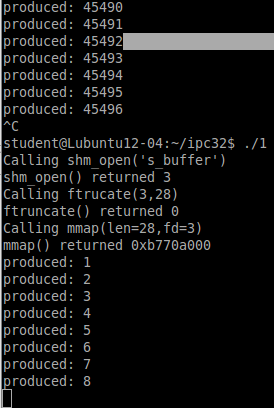
# Opdracht S

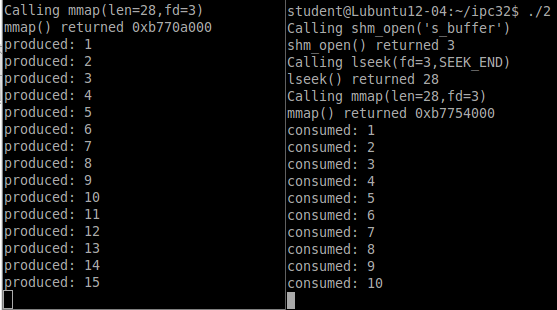
Joery van den Hoff

Semafoor s is niet nodig omdat de andere twee semaforen ervoor zorgen dat er nooit geprobeerd wordt om een waarde te schrijven / lezen die gebruikt wordt op een andere thread. (Gecombineerd met het feit dat wij maar één producer en één consumer hebben, voor de extra opdracht is hij uiteraard wel nodig.)

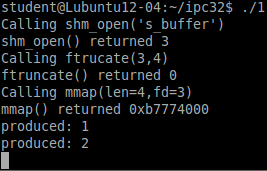
Hieronder heeft de producer de buffer volgeschreven en is aan het wachten op de consumer.

****

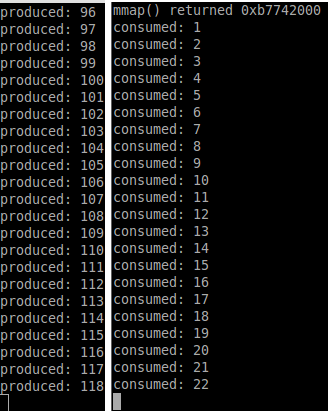
Hier kan je zien dat de producer en consumer aan het werken zijn.

****

Hier is de buffersize 1 en werken mijn programma’s nog steeds.

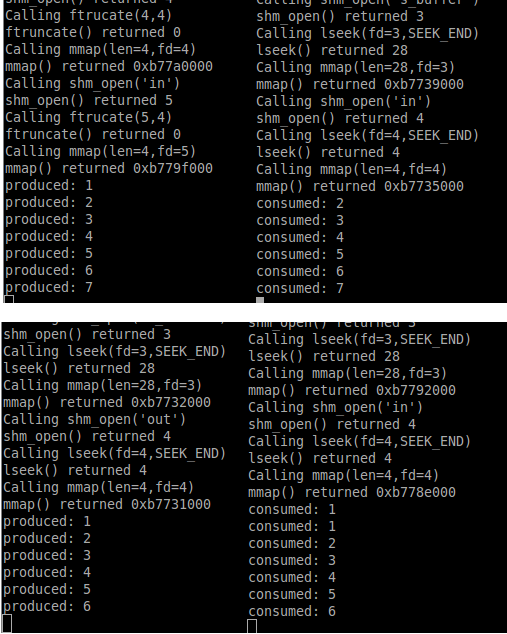
****

Hier is de buffersize 100, en mijn programma’s doen het nog steeds.

****

## Extra

Hier zie je 2 producers en 2 consumers tegelijk draaien.



* Dit heb ik gerealiseerd door de producer processen tijdelijk even uit te zetten.
* Dit heb ik gerealiseerd door de consumer processen tijdelijk even uit te zetten.
* Ik heb de usleep() aangepast in producer zodat hij sneller gaat dan consumer.
* Ik heb de usleep() aangepast in consumer zodat hij sneller gaat dan producer.

Alle situaties zijn zoals verwacht. Producer processen wachten tot ze weer ruimte hebben om te producen. Consumer processen wachten tot er weer iets te consumeren valt.