ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

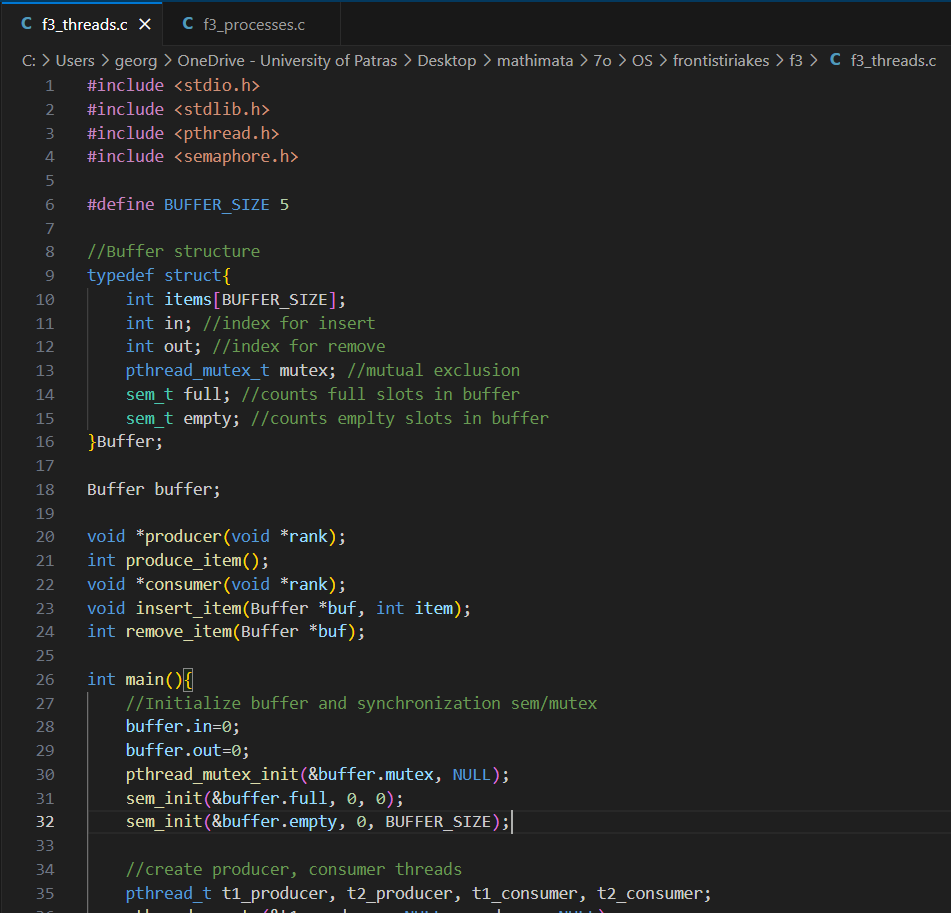
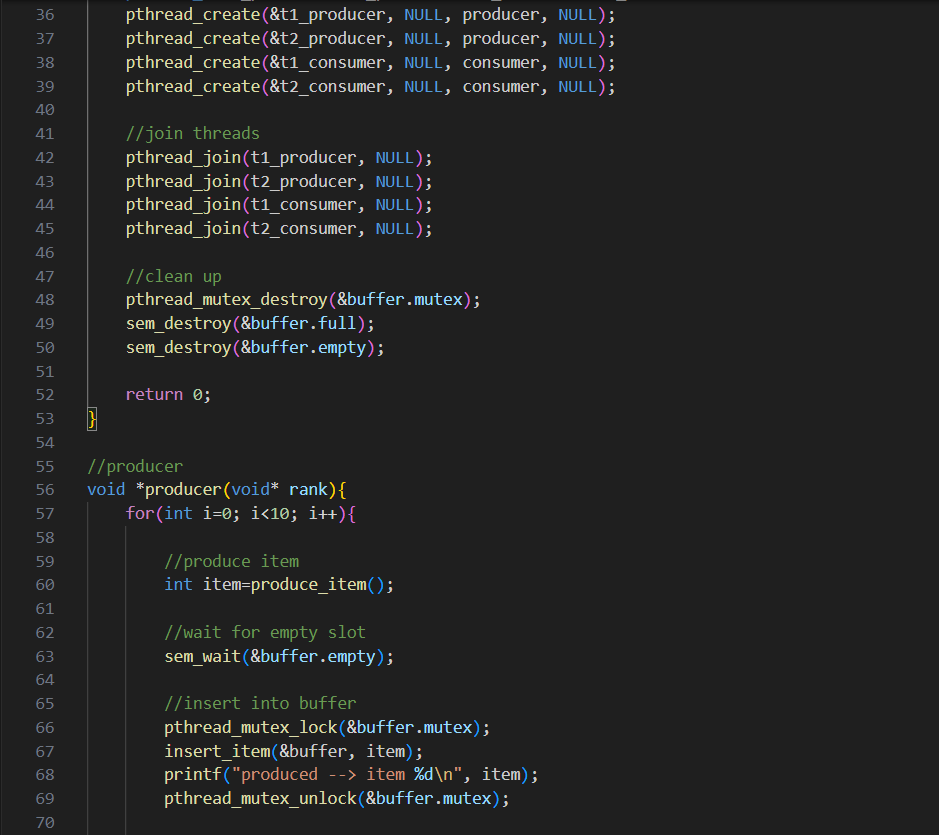
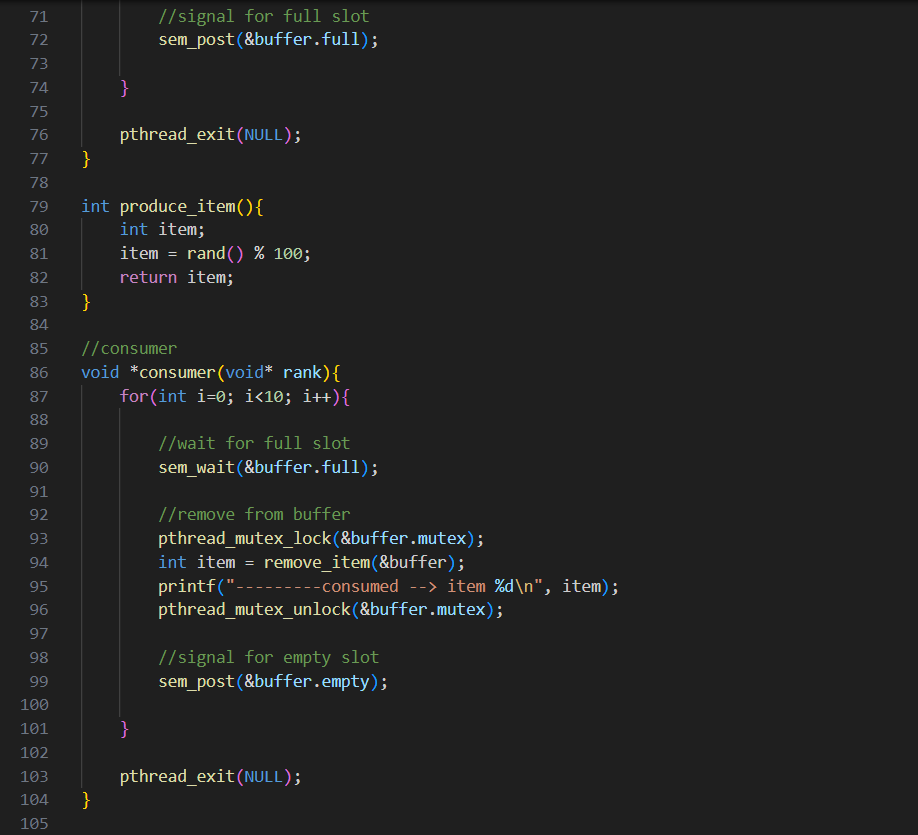
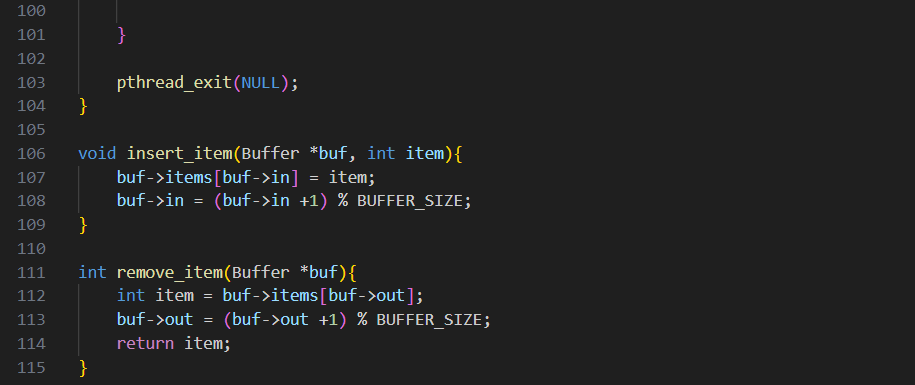
ΦΡΟΝΤΙΣΤΗΡΙΟ – ΑΣΚΗΣΗ 3

**ΜΠΑΣΑΓΙΑΝΝΗ ΓΕΩΡΓΙΑ**

**1084016**

Για την εκπόνηση της άσκησης τα αρχεία με τους κώδικες στη c γράφτηκαν με χρήση VSCode και για την μεταγλώττιση και εκτέλεση τους χρησιμοποιήθηκε το WSL (μέσω terminal).

# Το πρόβλημα του παραγωγού-καταναλωτή με νήματα

  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
ψς  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Ορίζουμε την χωρητικότητα του buffer να είναι 5 (πράγμα που μπορούμε να αλλάξουμε με μία και μόνο αλλαγή στο #define. Έπειτα, δημιουργούμε την δομή του buffer με τα προϊόντα items που παράγονται και καταναλώνονται, 2 indexes που ορίζουν την θέση που εισάγεται και που αφαιρείται κάποιο προϊόν από το buffer, το mutex του αμοιβαίου αποκλεισμού και δύο σημαφόρους για τα γεμάτα και άδεια slots κάθε φορά.

Αρχικοποιούνται τα in και out στην αρχή του buffer (θέση 0), το mutex με NULL και full=0, empty=BUFFER\_SIZE(=5). Δημιουργία των threads με την pthread\_create 2 για παραγωγή και 2 για κατανάλωση. Επικοινωνία των threads με την pthread\_join. Τέλος, καταστρέφει το mutex και τις σημαφόρους αφού πλέον δεν τις χρειάζεται.

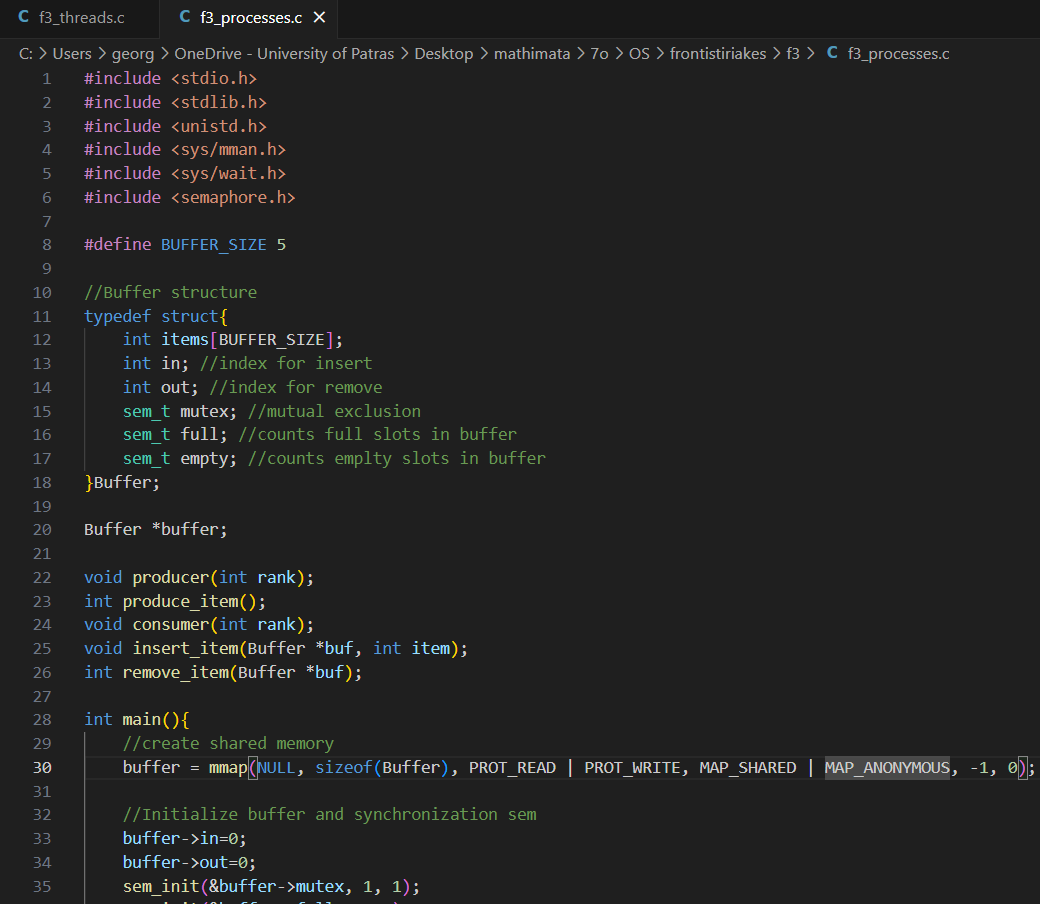
Τα threads που κάνουν τη δουλειά του producer δημιουργούν ένα item μέσω της produce\_item() που επιστρέφει έναν τυχαίο αριθμό μέσω της rand(), και κάνει wait\_empty και αν υπάρχει άδειο slot μπαίνει σε mutex και εισάγει με την insert\_item δεδομένο στο buffer. Επίσης μέσα στο mutex τυπώνει και το προϊόν που παράχθηκε (για τον κατάλληλο έλεγχο). Μετά, κάνει unlock το mutex και δίνει σήμα στην full.

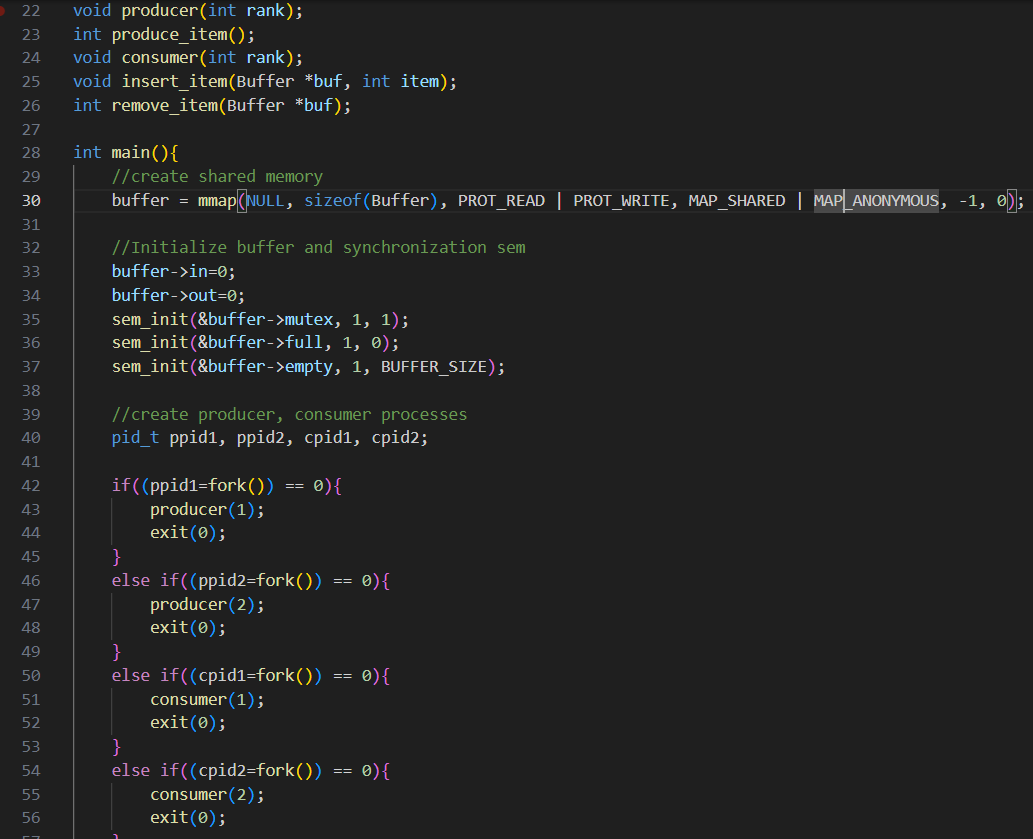
Τα threads που κάνουν τη δουλειά του consumer κάνουν wait\_full και αν υπάρχει γεμισμένο slot μπαίνει σε mutex και αφαιρεί ένα item από το buffer με την remove\_item. Επίσης μέσα στο mutex τυπώνει και το προϊόν που καταναλώθηκε (για τον κατάλληλο έλεγχο). Μετά, κάνει unlock το mutex και δίνει σήμα στην empty.

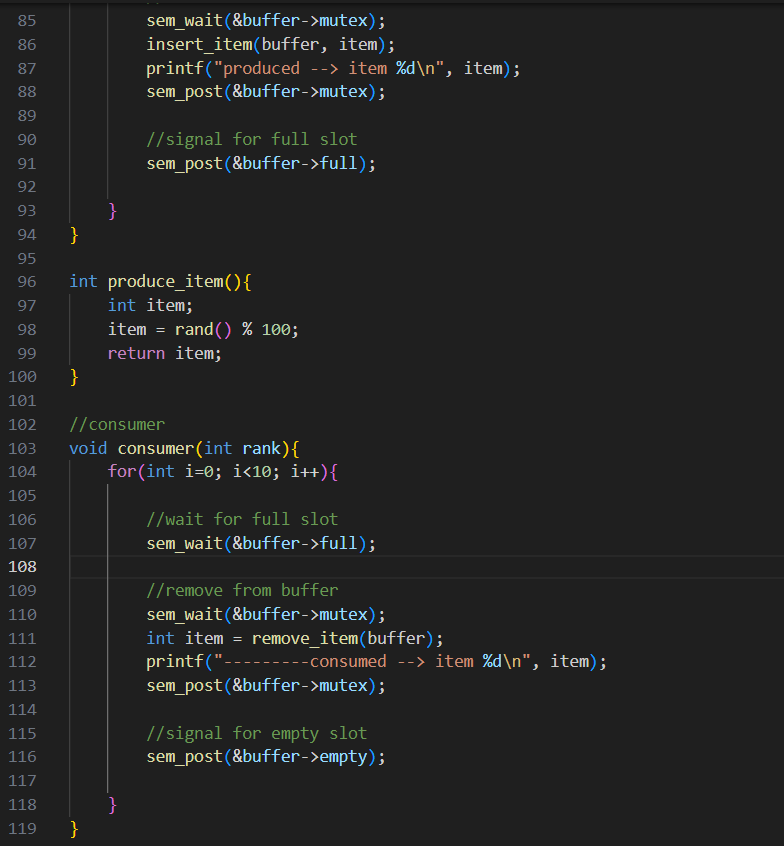
H insert\_item βάζει το παραγόμενο item στην θέση in του buffer και αυξάνει την θέση in κατά 1 για να δείχνει στην επόμενη θέση. Η remove\_item παίρνει το item της θέσης out του buffer και το επιστρέφει, αφού πρώτα αυξήσει το out κατά 1, στην επομένη θέση του buffer. Για να λειτουργήσει το +1 πρέπει να χρησιμοποιηθεί και το % modulo έτσι ώστε οι τιμές των in και out να είναι από 0-4 δηλαδή μέσα στα πλαίσια του BUFFER\_SIZE.

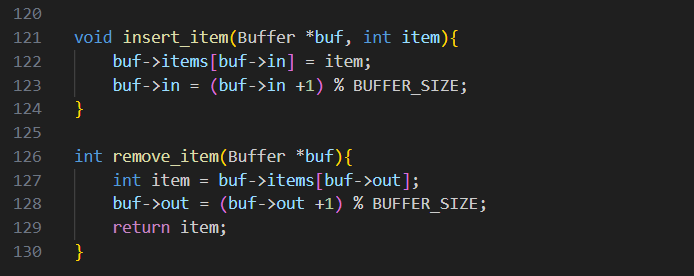
## Αποτελέσματα:

# Το πρόβλημα του παραγωγού-καταναλωτή με διεργασίες





  
σ

  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
χς

Η λογική είναι η ίδια με αυτή που επιγράφθηκε στη λύση με τα νήματα.

Αξίζει να σημειωθεί ότι τώρα που πρόκειται για διεργασίες η κοινή μνήμη του buffer ορίζεται στην main με την mmap με δικαιώματα γραφής και ανάγνωσης από όλες τις διεργασίες. Στο τέλος του προγράμματος με την munmap αποδεσμεύεται η κοινή μνήμη που είχε δεσμευτεί.

## Αποτελέσματα:

