

Διάλεξη 13: Κατανεμημένη Κοινόχρηστη Μνήμη

ΕΠΛ 432: Κατανεμημένοι Αλγόριθμοι



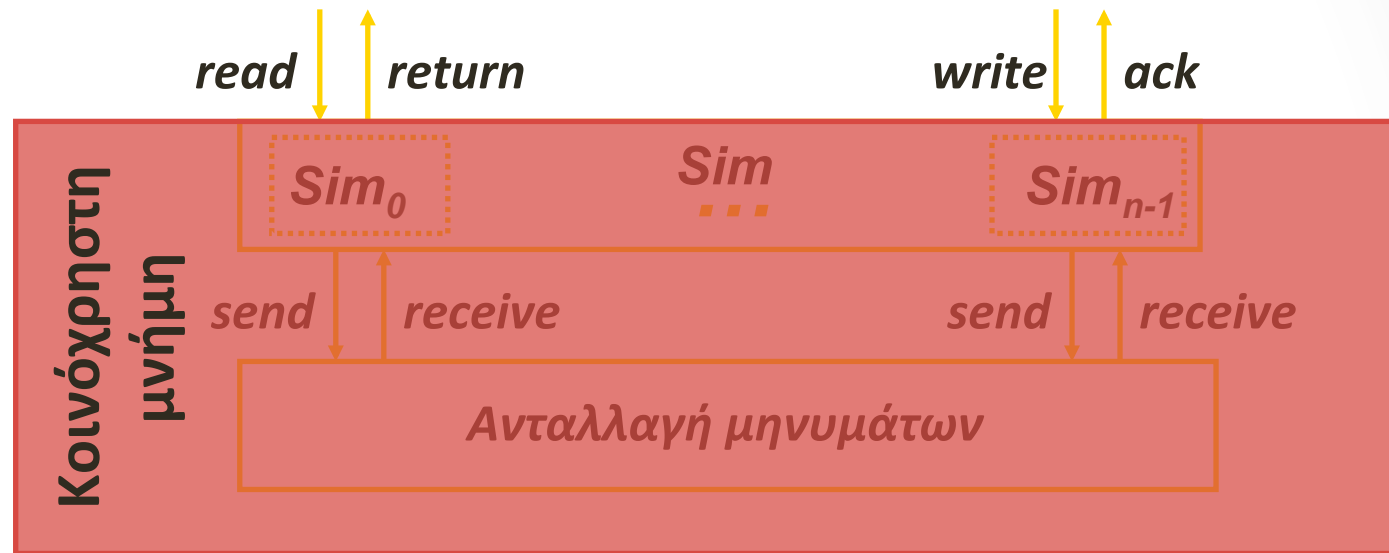
Τι θα δούμε σήμερα

- Προσομοίωση Κοινόχρηστης Μνήμης
- Συνθήκες Συνέπειας
- Αλγόριθμος χωρίς σφάλματα

Κατανεμημένη Κοινόχρηστη Μνήμη

- Προσφέρει την ψευδαίσθηση μιας **κοινόχρηστης μεταβλητής** πάνω από ένα μοντέλο ανταλλαγής μηνυμάτων
- Γιατί χρειαζόμαστε την ΚΚΜ;
 - Πολλές φορές είναι πιο εύκολο να σχεδιάσουμε αλγορίθμους και να γράψουμε προγράμματα στην κοινόχρηστη μνήμη παρά στο περιβάλλον ανταλλαγής μηνυμάτων
 - Ανοχή σε σφάλματα
- Στόχος: Να **προσομοιώσουμε** το επικοινωνιακό μοντέλο της κοινόχρηστης μνήμης πάνω από το επικοινωνιακό μοντέλο ανταλλαγής μηνυμάτων
 - Θέλουμε να προσομοιώσουμε **read/write μεταβλητές**

Προσομοίωση Κοινόχρηστης Μνήμης



Προσομοιώνει



Διεργασίες και Λειτουργίες στη ΚΚΜ

- Μια διεργασία μπορεί να εκτελέσει δύο λειτουργίες πάνω στην ΚΚΜ: **Ανάγνωση και Εγγραφή**
- Οι δύο αυτές λειτουργίες αποτελούνται από δύο μέρη:
 - **Αίτηση και Απάντηση**
- Στάδια εκτέλεσης μιας λειτουργίας
 - Αίτηση ανάγνωσης/εγγραφής
 - Ανταλλαγή μηνυμάτων για διεκπεραίωση της αίτησης
 - Απάντηση για την ανάγνωση/εγγραφή

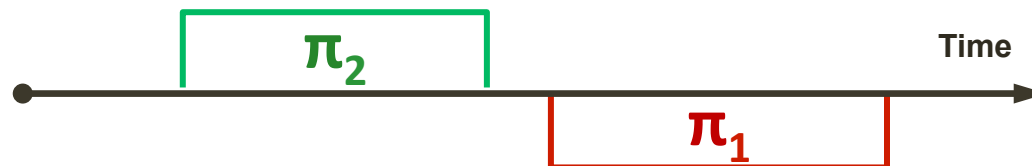
Οι λειτουργίες δεν διεκπεραιώνονται στιγμιαία!!

Σχέση Προτεραιότητας Λειτουργιών

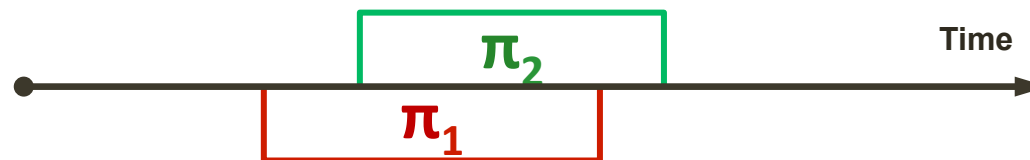
- Οι σχέσεις προτεραιότητας μεταξύ δύο λειτουργιών π_1 , π_2 :
 - Η π_1 **προηγείται** της π_2 πάνω στην ΚΚΜ εάν η απάντηση της π_1 συμβαίνει χρονικά **πριν-από** την αίτηση της π_2



- Η π_1 **έπεται** της π_2 πάνω στην ΚΚΜ εάν η αίτηση της π_1 συμβαίνει **μετά-από** την απάντηση της π_2



- Η π_1 είναι **ταυτόχρονη** με την π_2 εάν η π_1 **ούτε προηγείται και ούτε έπεται** της π_2



Μοντέλα Ταυτοχρονισμού ΚΚΜ

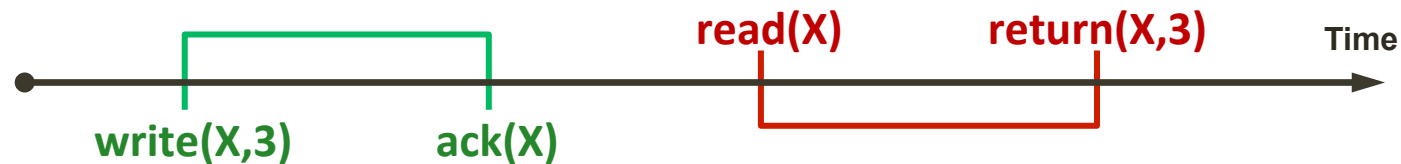
- Μοντέλο **ενός εγγραφέα** και **ενός αναγνώστη** (SWSR)
 - Μια εγγραφή ταυτόχρονη με μια ανάγνωση πάνω σε κάποια μεταβλητή X της ΚΚΜ
- Μοντέλο **ενός εγγραφέα** και **πολλών αναγνωστών** (SWMR)
 - Μια εγγραφή ταυτόχρονη με πολλές ταυτόχρονες αναγνώσεις πάνω σε κάποια μεταβλητή X της ΚΚΜ
- Μοντέλο **πολλών εγγραφέων** και **πολλών αναγνωστών** (MWMR)
 - Πολλές ταυτόχρονες εγγραφές ταυτόχρονες με πολλές ταυτόχρονες αναγνώσεις πάνω σε κάποια μεταβλητή X της ΚΚΜ

Ακολουθιακή Συμπεριφορά Λειτουργιών

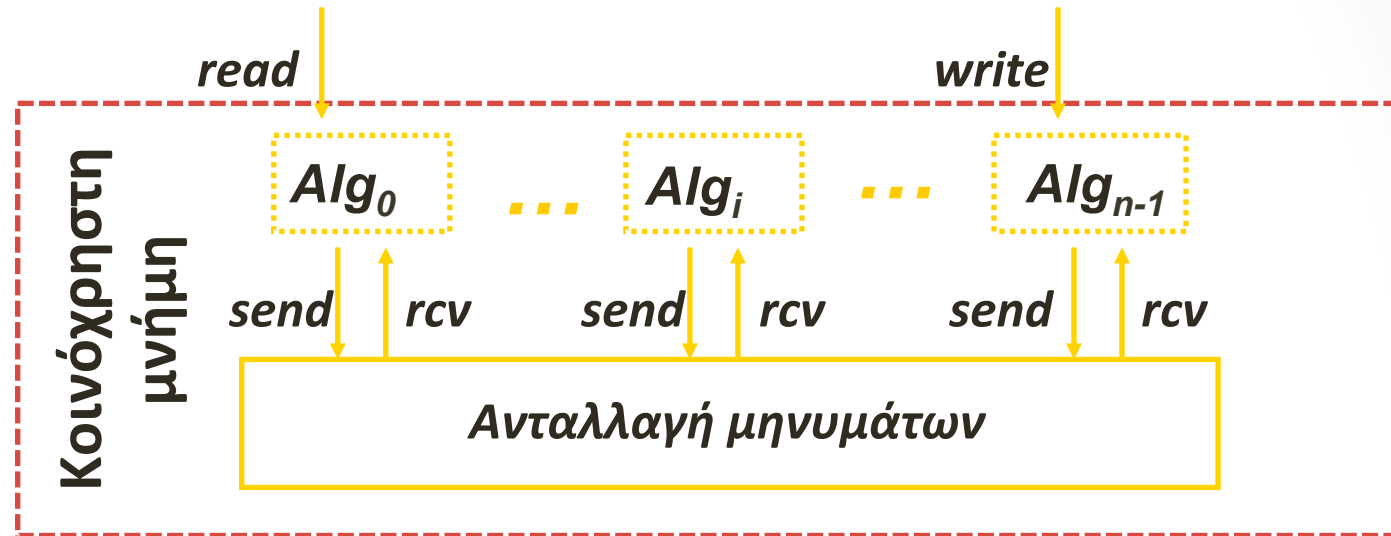
- Ακολουθιακή Συμπεριφορά: Συμπεριφορά ΚΚΜ στην **απουσία ταυτοχρονίας** μεταξύ των λειτουργιών
- Ανάγνωση
 - Αίτηση: `read(X)`
 - Απάντηση: `return(X,v)`
- Εγγραφή
 - Αίτηση: `write(X,v)`
 - Απάντηση: `ack(X)`
- Κάθε αίτηση έχει μια αντίστοιχη απάντηση

Ακολουθιακή Συμπεριφορά Λειτουργιών

- Μια **ακολουθία λειτουργιών είναι νόμιμη** εαν και μόνο εαν κάθε ανάγνωση επιστρέφει την τιμή που γράφτηκε από την τελευταία εγγραφή που **προηγείται**



Ταυτοχρονία πάνω στην ΚΚΜ



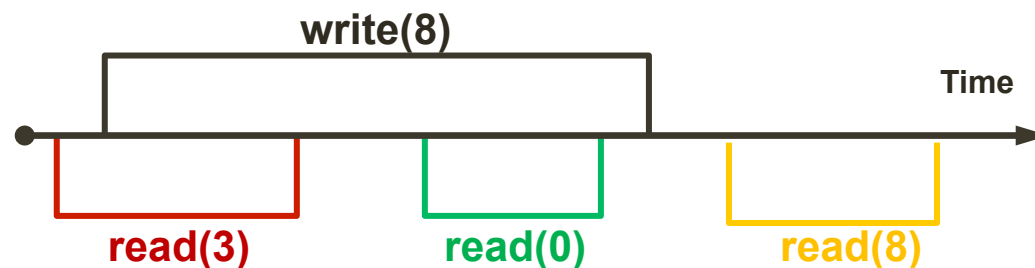
- Ταυτοχρονία: Λειτουργίες καλύπτονται χρονικά
 - Αφού δεν είναι στιγμιαίες
- Επομένως μια ανάγνωση μπορεί να συμβαίνει ταυτόχρονα με μια εγγραφή
 - Τι τιμή πρέπει να επιστρέψει η ανάγνωση;
- Καθορίζεται από τη **Συνθήκη Συνέπειας της Μνήμης**

Συνθήκες Συνέπειας

- Συνδυάζουν την ακολουθιακή συμπεριφορά με το καθορισμό του τι συμβαίνει στην παρουσία ταυτοχρονισμού
- Θα μελετήσουμε τέσσερις συνθήκες συνέπειας
 - Ασφαλής Συνέπεια
 - Κανονικότητα
 - Ατομικότητα (Γραμμικοποιησιμότητα)
 - Ακολουθιακή

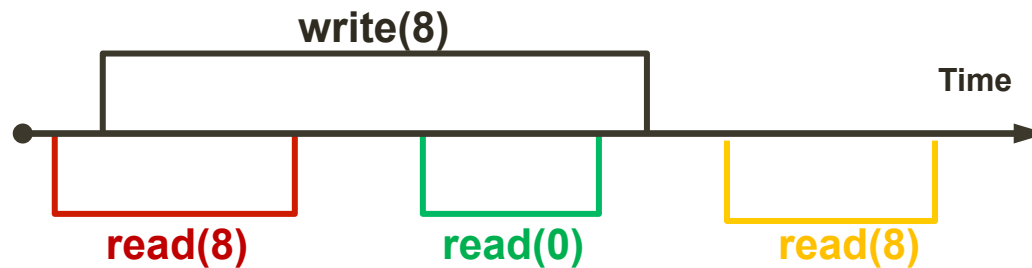
Ασφαλής Συνέπεια

- Εάν μια λειτουργία ανάγνωσης δεν είναι ταυτόχρονη με μια λειτουργία εγγραφής τότε η ανάγνωση επιστρέφει την τιμή που γράφτηκε από την τελευταία εγγραφή που **προηγείται**
 - Ικανοποιεί την **Νόμιμη ακολουθία λειτουργιών σε κάθε μεταβλητή**
- Καμιά εγγύηση δεν δίδετε για τις τιμές που επιστρέφουν οι αναγνώσεις που είναι ταυτόχρονες με εγγραφές



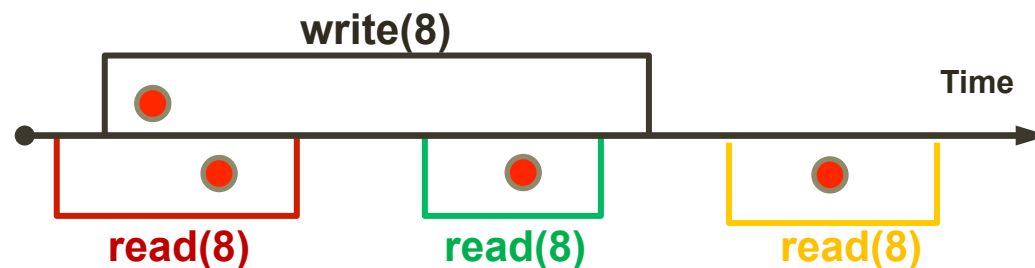
Κανονικότητας

- Ικανοποιεί την Ασφαλή Συνέπεια και επιπλέον...
- Εάν μια ανάγνωση είναι ταυτόχρονη με μια εγγραφή τότε η ανάγνωση επιστρέφει
 - είτε την τιμή της αμέσως προηγούμενης εγγραφής
 - είτε την τιμή που αναγράφεται από την ταυτόχρονη εγγραφή



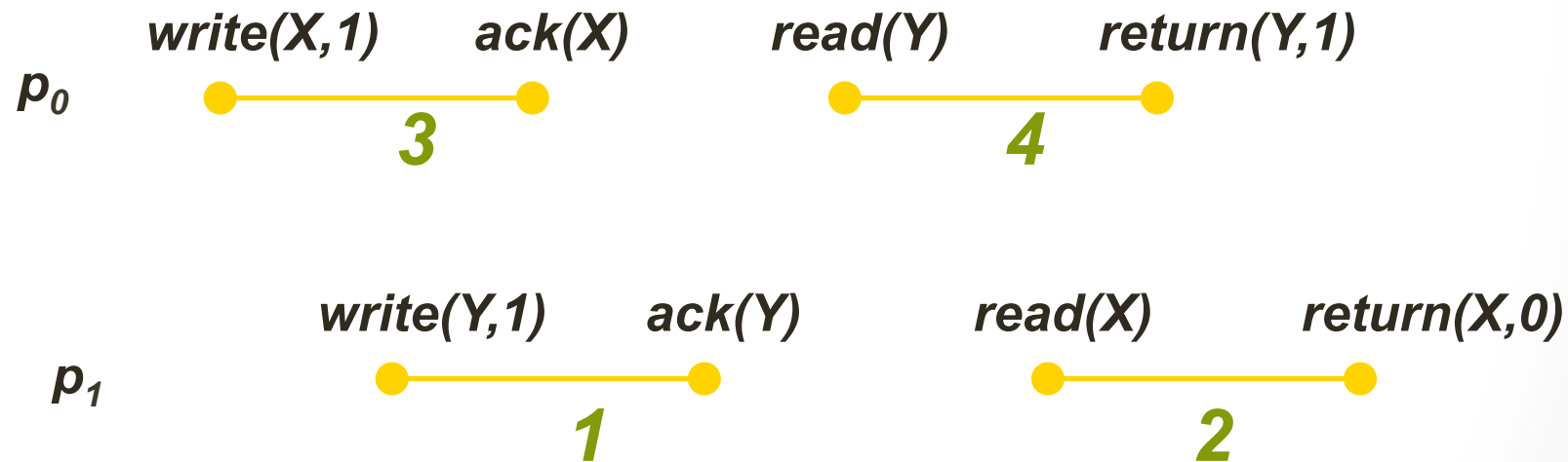
Ατομικότητας

- Ικανοποιεί την κανονικότητα και επιπλέον...
- Ικανοποιείται η καθολική χρονική σειρά των λειτουργιών
- Μπορούμε να συρρικνώσουμε κάθε λειτουργία σε ένα χρονικό σημείο και να προσφέρουμε την ψευδαίσθηση μιας ακολουθιακής εκτέλεσης



Ακολουθιακή Συνέπεια

- Ικανοποιεί την Νόμιμη ακολουθία λειτουργιών σε κάθε μεταβλητή
- Ικανοποιεί την χρονική σειρά των λειτουργιών σε κάθε επεξεργαστή

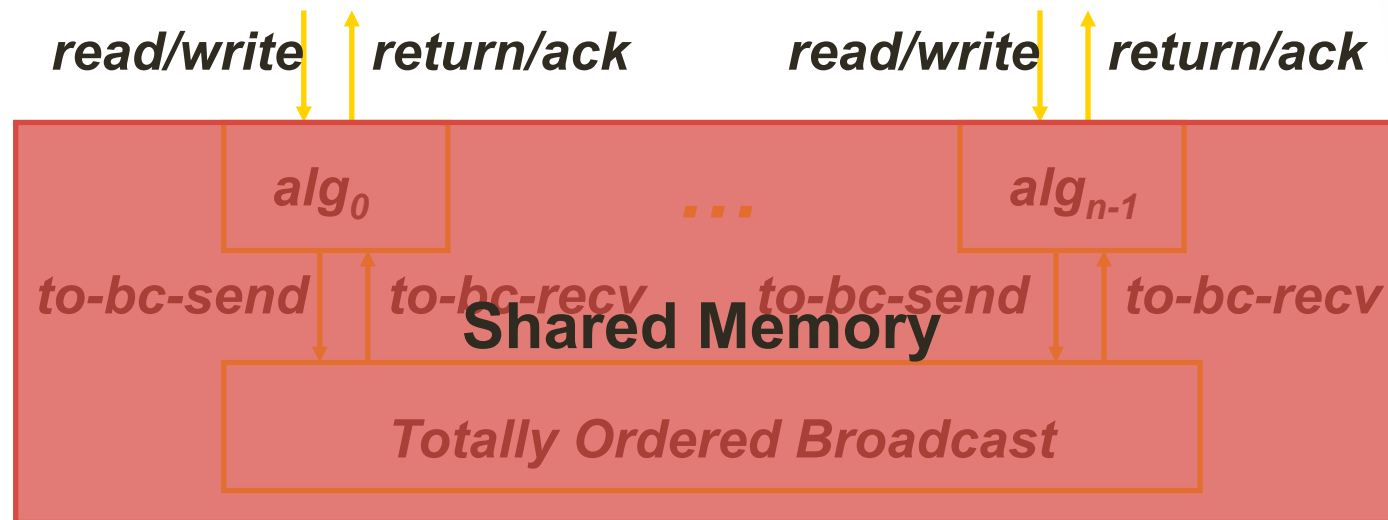


Αλγόριθμος για Ατομική ΚΚΜ

- Χρησιμοποίηση υπηρεσίας ολικής διάταξης ως το κατώτερο σύστημα επικοινωνίας.
- Κάθε επεξεργαστής κρατά ένα αντίτυπο της κάθε κοινόχρηστης μεταβλητής
- Όταν παραλάβουμε μια αίτηση ανάγνωσης:
 - Στείλε μήνυμα διάχυσης με την αίτηση ανάγνωσης
 - Όταν λάβεις το δικό σου μήνυμα, επέστρεψε τη τιμή της τοπικής σου μεταβλητής
- Όταν παραλάβουμε μια αίτηση εγγραφής:
 - Στείλε μήνυμα διάχυσης με την αίτηση εγγραφής
 - Όταν λάβεις αίτηση εγγραφής, ενημέρωσε την τοπική σου μεταβλητή
 - Όταν λάβεις το δικό σου μήνυμα, επέστρεψε ack

Προσομοίωση

users of read/write shared memory



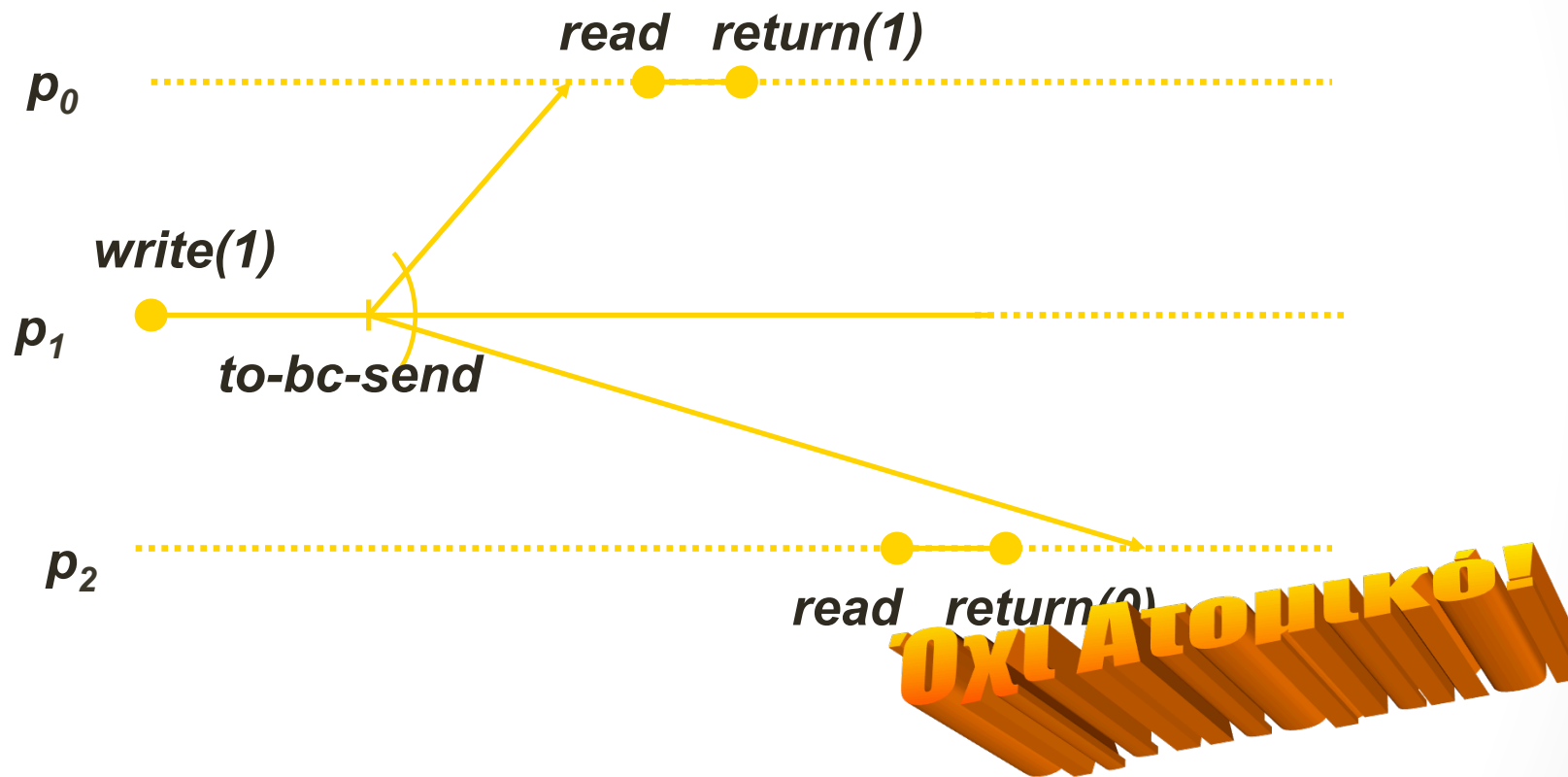
Ορθότητα Αλγορίθμου

- Υποθέστε οποιαδήποτε νόμιμη εκτέλεση α του αλγορίθμου όπου:
 - Η υπηρεσία ολικά διατεταγμένης διάχυσης λειτουργεί ορθά
 - Κάθε χρήστης περιμένει να τελειώσει κάποια αίτησή του πριν κάνει την επόμενη αίτηση (μια αίτηση τη φορά)
- Πρέπει να δείξουμε ότι τα γεγονότα που εμφανίζονται στην α ικανοποιούν:
 - Συνθήκη Ζωτικότητας: Κάθε αίτηση έχει απάντηση και
 - Συνθήκη Ασφαλείας: Ατομικότητα (Γραμμικοποιησιμότητα)

Correctness of Linearizability Algorithm

- **Ζωτικότητα:**
 - Διασφαλίζεται από την ζωτικότητα της ολικά διατεταγμένης διάχυσης
- **Ατομικότητα:** Ορίζουμε την διάταξη π των λειτουργιών ως η σειρά με την οποία παραλήφθηκαν τα αντίστοιχα μηνύματα διάχυσης
 - π είναι νόμιμη: αφού όλες οι λειτουργίες ακολουθούν τη σειρά του TO bcast.
 - π ικανοποιεί χρονική σειρά λειτουργιών: αν O_1 τελειώσει πριν το O_2 ξεκινήσει, το μήνυμα του O_1 ταξινομείται πριν από το μήνυμα του O_2 .

Γιατί χρειάζεται η διάχυση από την ανάγνωση;



Ερωτήσεις;

