

Διάλεξη 1: Εισαγωγή στον Κατανεμημένο Υπολογισμό

ΕΠΛ 432: Κατανεμημένοι Αλγόριθμοι



Τι θα δούμε σήμερα

- Τι είναι ένα Κατανεμημένο Σύστημα;
- Επικοινωνία, Χρονισμός, Σφάλματα
- Μοντέλο Ανταλλαγής Μηνυμάτων

Κατανεμημένα Συστήματα

- **Κατανεμημένο Σύστημα:** Μια συλλογή υπολογιστικών μονάδων που επικοινωνούν μεταξύ τους.

«Κατανεμημένο Σύστημα είναι αυτό στο οποίο η κατάρρευση κάποιου υπολογιστή που δεν ξέραμε ότι υπήρχε μπορεί να θέσει το σύστημά μας εκτός λειτουργίας»

Leslie Lamport

«Κατανεμημένο Σύστημα είναι μια συλλογή από ανεξάρτητες υπολογιστικές μονάδες που παρουσιάζονται στον χρήστη ως μια μοναδική οντότητα»

Tanenbaum, van Steen

Παράμετροι

- Μοντέλα Επικοινωνίας
- Μοντέλα Χρονισμού
- Σφάλματα
 - Τύποι
 - Αριθμός

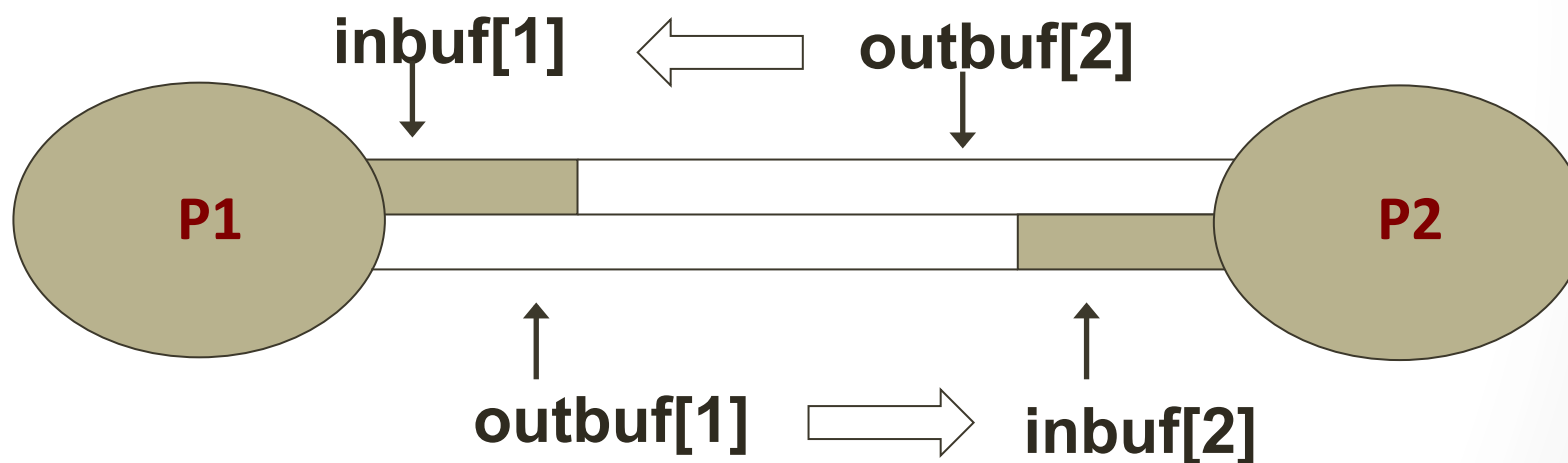
Επικοινωνία

- Κοινόχρηστη Μνήμη
 - Οι επεξεργαστές επικοινωνούν μέσω μιας κοινής μνήμης που περιλαμβάνει ένα σύνολο από κοινές μεταβλητές.
 - Τύποι Μεταβλητών
 - Ανάγνωσης και Γραφής (read/write)
 - Ανάγνωσης-Μετατροπής-Γραφής (read-modify-write)
 - Ελέγχου και Αλλαγής (test&set)
 - Σύγκρισης και Αντιμετάθεσης (compare&swap)



Επικοινωνία

- Ανταλλαγή Μηνυμάτων
 - Οι επεξεργαστές επικοινωνούν εσωκλείοντας πληροφορία σε μηνύματα που στέλνονται στο δίκτυο
 - Κάθε επεξεργαστής διαθέτει
 - Λίστα εισερχόμενων μηνυμάτων
 - Λίστα εξερχόμενων μηνυμάτων



Χρονισμός

- Σύγχρονο Μοντέλο

- Συγχρονισμένα βήματα επεξεργαστών
- Ο υπολογισμός προχωράει σε γύρους
- Υπάρχει γνωστό όριο τερματισμού κάποιου γύρου και όλα τα μηνύματα παραδίδονται με το τέλος του γύρου

- Ασύγχρονο Μοντέλο

- Αυθαίρετες διαφορές στις ταχύτητες των επεξεργαστών
- Αυθαίρετες καθυστερήσεις μηνυμάτων / Αυθαίρετη καθυστέρηση πρόσβασης της κοινόχρηστης μνήμης

Τι θα μελετήσουμε;

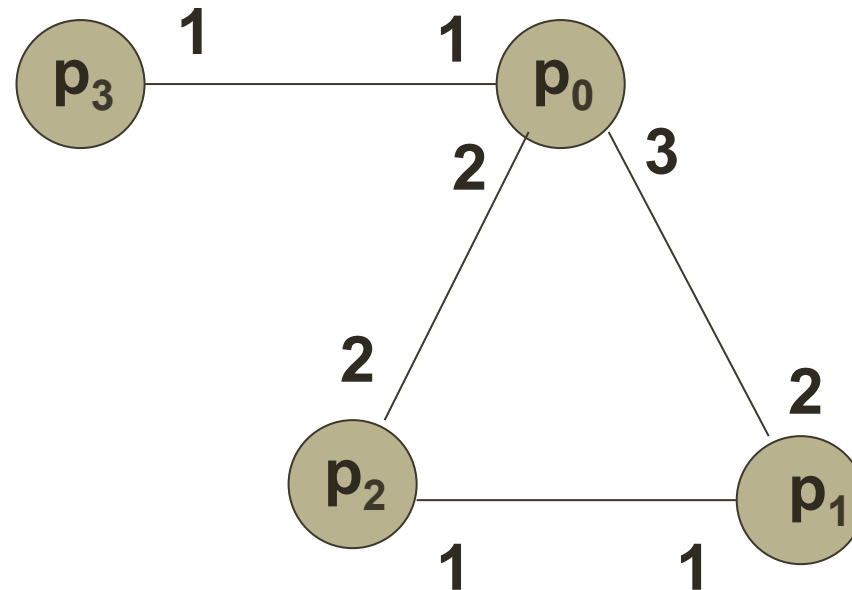
	Ανταλλαγής Μηνυμάτων	Κοινόχρηστης Μνήμης
Σύγχρονο	Ναι	Όχι
Ασύγχρονο	Ναι	Ναι

(Σύγχρονο Μοντέλο Κοινόχρηστης Μνήμης: PRAM)

Σφάλματα

- Περιγράφουν τα προβλήματα που μπορούν να αντιμετωπίσουν οι επεξεργαστές του συστήματος
- **Σφάλματα Κατάρρευσης**
 - Ένας επεξεργαστής σταματά να λαμβάνει μέρος στον υπολογισμό.
 - Ιδανικό σενάριο σε πραγματικά συστήματα
- **Βυζαντινά Σφάλματα**
 - Αυθαίρετη συμπεριφορά επεξεργαστών
 - Χρησιμοποιούνται όταν τα σφάλματα των επεξεργαστών είναι άγνωστα ή κακόβουλα
- Ένας καταναεμημένος αλγόριθμος σχεδιάζεται με βάση τον **αριθμό και τον τύπο σφαλμάτων** που μπορεί να ανεχτεί

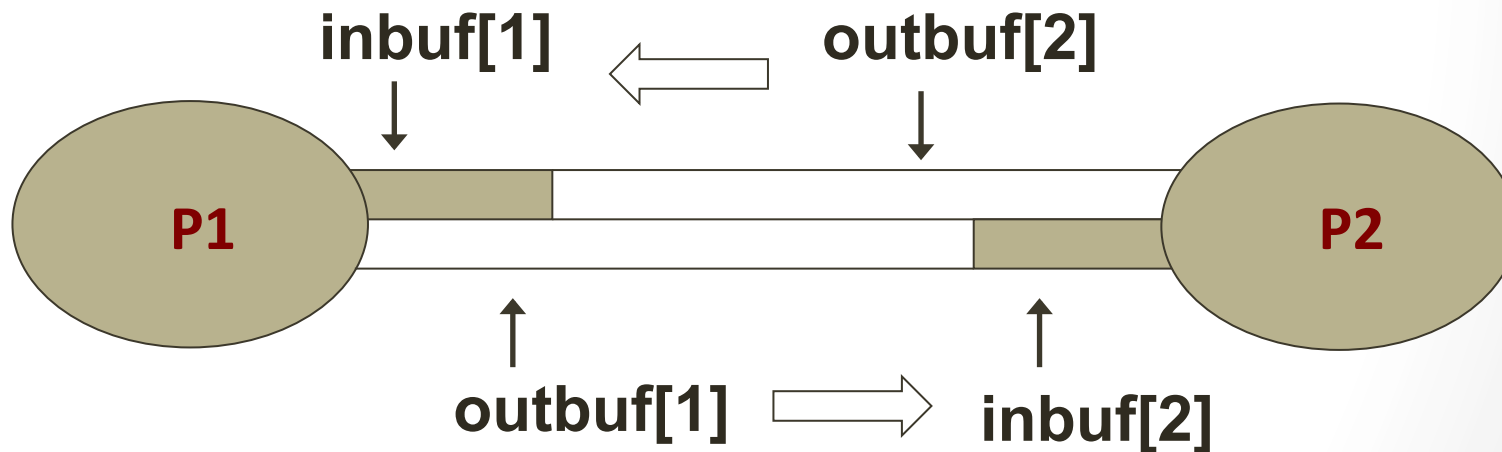
Μοντέλο Ανταλλαγής Μηνυμάτων



- Οι επεξεργαστές p_0, p_1, \dots, p_{n-1} είναι οι κόμβοι του γραφήματος
- Κανάλια επικοινωνίας σημείο-προς-σημείο, διπλής κατεύθυνσης (μη κατευθυνόμενες ακμές γράφου)
- Κάθε επεξεργαστής
 - Γνωρίζει και μπορεί να αριθμήσει τα εξερχόμενα κανάλια
 - Μπορεί να μην γνωρίζει ποιος επεξεργαστής βρίσκεται στο άλλο άκρο

Επεξεργαστές και Κανάλια

- Ένας επεξεργαστής είναι μια **μηχανή καταστάσεων**
 - Δυνατό να περιλαμβάνει άπειρο αριθμό καταστάσεων
- **Κανάλι Επικοινωνίας**
 - Προς-τα-έξω/εξερχόμενο καταχωρητής (φυσικό κανάλι)
 - Προς-τα-έσω/εισερχόμενο καταχωρητής (λίστα εισερχόμενων μηνυμάτων)



Διατάξεις

- Κατάσταση Επεξεργαστή
 - Η συλλογή των τοπικών **μεταβλητών κατάστασης** (τιμές τοπικών μεταβλητών του επεξεργαστή)
 - Προς-τα-έσω καταχωρητής
- Κατάσταση Συστήματος (**Διάταξη**)
 - Το **διάνυσμα των καταστάσεων των επεξεργαστών**
 - Συλλαμβάνει την τρέχουσα εικόνα του συστήματος
 - Καταστάσεις Επεξεργαστών
 - Καταστάσεις Καναλιών Επικοινωνίας

Γεγονότα

- Παραλαβή μηνύματος ($rcv(M)$)
 - Μετακίνηση μηνύματος από τον **εξερχόμενο καταχωρητή αποστολέα** στον **εισερχόμενο καταχωρητή του παραλήπτη**
 - Παραλήπτης επεξεργάζεται το μήνυμα στο επόμενο υπολογιστικό του βήμα
- Υπολογιστικό Βήμα $\langle state1, action, state2 \rangle$ ενός επεξεργαστή p_i
 - $state1$: μια προσβάσιμη κατάσταση του p_i
 - Τοπικές μεταβλητές και εισερχόμενα μηνύματα
 - $action$: συνάρτηση μετάβασης που εφαρμόζεται πάνω στο $state$
 - Επεξεργασία εισερχόμενων μηνυμάτων
 - $state2$: τελική κατάσταση του p_i μετά την εφαρμογή της μετάβασης $action$ στην κατάσταση $state$
 - Εισερχόμενος καταχωρητής κενός
 - Εξερχόμενος καταχωρητής περιέχει τυχών μηνύματα που θα σταλούν

Ασύγχρονες Εκτελέσεις

- Αλληλουχία:
 - διάταξη, γεγονός, διάταξη, γεγονός,..., διάταξη, γεγονός,...
- Αρχική Διάταξη
 - Κάθε επεξεργαστής βρίσκεται στην αρχική του κατάσταση
 - Όλοι οι εισερχόμενοι καταχωρητές είναι κενοί
- Για κάθε τριάδα <διάταξη, γεγονός, διάταξη>, η **νέα διάταξη είναι ίδια με την παλιά διάταξη** εκτός από τις εξής διαφορές:
 - **Παραλαβή Μηνύματος:** το μήνυμα που παραλαμβάνεται μετακινείται από τον εξερχόμενο καταχωρητή του αποστολέα στον εισερχόμενο καταχωρητή του παραλήπτη
 - **Υπολογισμός:** η κατάσταση του επεξεργαστή που εκτελεί το υπολογιστικό βήμα (μαζί με τον εισερχόμενο καταχωρητή) αλλάζει σύμφωνα με την συνάρτηση μετάβασης του.

Νόμιμες Εκτελέσεις (Συνθήκες Ορθότητας)

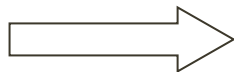
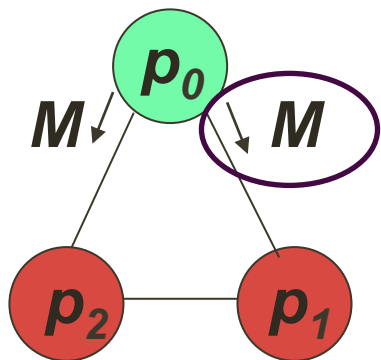
- Συνθήκες που πρέπει να ικανοποιούνται
 - **Ασφάλειας:** τίποτα κακό δεν συμβαίνει
 - Διασφαλίζει ότι η εκτέλεση ακολουθεί σωστά τις μεταβάσεις
 - **Ζωτικότητας:** κάτι καλό θα συμβεί
 - Διασφαλίζει ότι κάποια πράγματα θα συμβούν και το σύστημα δεν θα μείνει άπρακτο επ' άπειρον
- Συνθήκες ζωτικότητας για το ασύγχρονο μοντέλο
 - Κάθε μήνυμα σε ένα εξερχόμενο καταχωρητή παραλαμβάνεται τελικά
 - Κάθε επεξεργαστής εκτελεί ένα άπειρο αριθμό βημάτων

Προσοχή: Δεν υπάρχει περιορισμός στο **πότε** συμβαίνουν αυτά τα γεγονότα

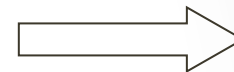
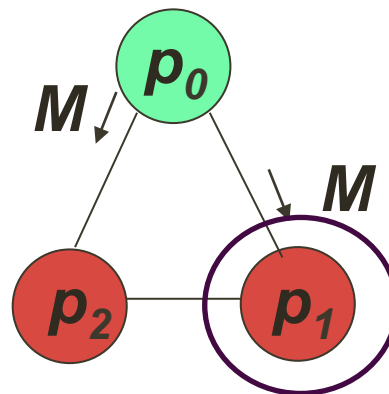
Παράδειγμα: Υπερχείλιση Μηνυμάτων

- Κατάσταση κάθε επεξεργαστή ορίζεται από
 - Τοπική μεταβλητή χρώμα με τιμές **πράσινο** ή **κόκκινο**
 - Εισερχόμενοι και εξερχόμενοι καταχωρητές μηνυμάτων
- Αρχικές καταστάσεις
 - p_0 : χρώμα=**πράσινο** και εξερχόμενοι καταχωρητές περιέχουν το μήνυμα M
 - υπόλοιποι: χρώμα=**κόκκινο** και εξερχόμενοι καταχωρητές κενοί
- Μετάβαση
 - Αν το M βρίσκεται στον εισερχόμενο καταχωρητή σου και το χρώμα=**κόκκινο**, αλλάξε το χρώμα σου σε **πράσινο** και στείλε M σε όλους τους εξερχόμενους καταχωρητές σου.

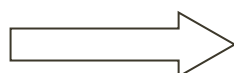
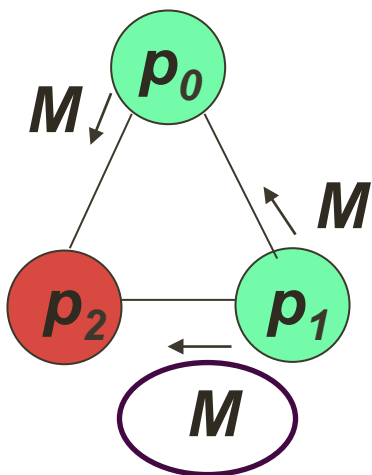
Παράδειγμα: Υπερχείλιση Μηνυμάτων



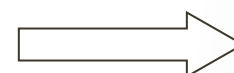
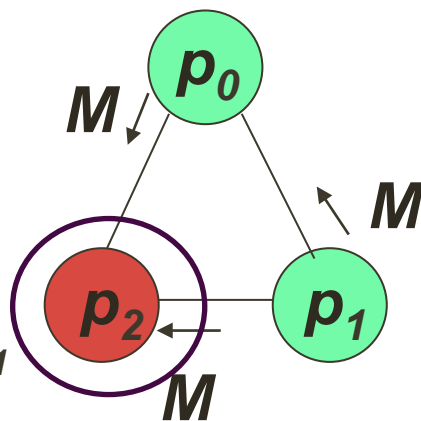
Γεγονός rcv()
στον p_1 από p_0



Υπολογισμός
από p_1



Γεγονός rcv()
στον p_2 από p_1



Υπολογισμός
από p_2 κτλ.
*Μέχρι να σταλούν
όλα τα μηνύματα*

Τερματισμός

- Οι νόμιμες εκτελέσεις ορίζονται ως άπειρες
- Συχνά όμως οι αλγόριθμοι **τερματίζουν**
- **Τερματικές Καταστάσεις Επεξεργαστών:**
 - Καταστάσεις από τις οποίες οι επεξεργαστές δεν μπορούν να φύγουν
- (Νομιμη) Εκτέλεση **Τερματίζει** αν:
 - Όλοι οι επεξεργαστές μπαίνουν σε **τερματικές καταστάσεις**
 - Δεν υπάρχουν μετεπιβιβαζόμενα μηνύματα

Πολυπλοκότητα

- **Πολυπλοκότητα Μηνυμάτων:** Μέγιστος αριθμός μηνυμάτων που αποστέλλονται σε οποιαδήποτε νόμιμη εκτέλεση
- **Χρονική Πολυπλοκότητα:** Μέγιστος «χρόνος» μέχρι τον τερματισμό σε οποιαδήποτε νόμιμη εκτέλεση
- Πως μετρούμε το χρόνο σε ένα ασύγχρονο περιβάλλον;

Χρονική Πολυπλοκότητα

- Κατασκευάζουμε μια χρονισμένη εκτέλεση αποδίδοντας μη-φθίνοντες χρόνους στα γεγονότα
 - Χρόνος μεταξύ αποστολής-παραλαβής μηνύματος είναι το πολύ 1
 - Υπολογιστικό βήμα αμέσως μετά την παραλαβή ενός μηνύματος
- Κανονικοποίηση Νόμιμης εκτέλεσης
 - Μέγιστη καθυστέρηση μηνύματος το πολύ 1 χρονική μονάδα
 - Επιτρέπει αυθαίρετες αναμείξεις γεγονότων
- **Χρονική Πολυπλοκότητα:** Μέγιστος «χρόνος» μέχρι τον τερματισμό σε οποιαδήποτε **χρονισμένη νόμιμη εκτέλεση**

Πολυπλοκότητα Αλγορίθμου

- Τερματική Κατάσταση
 - Το χρώμα του επεξεργαστή είναι πράσινο
- Πολυπλοκότητα Μηνυμάτων
 - $2m$, m είναι ο αριθμός των ακμών
- Χρονική Πολυπλοκότητα
 - $\text{diam}(G)+1$, όπου $\text{diam}(G)$ η διάμετρος του γράφου G

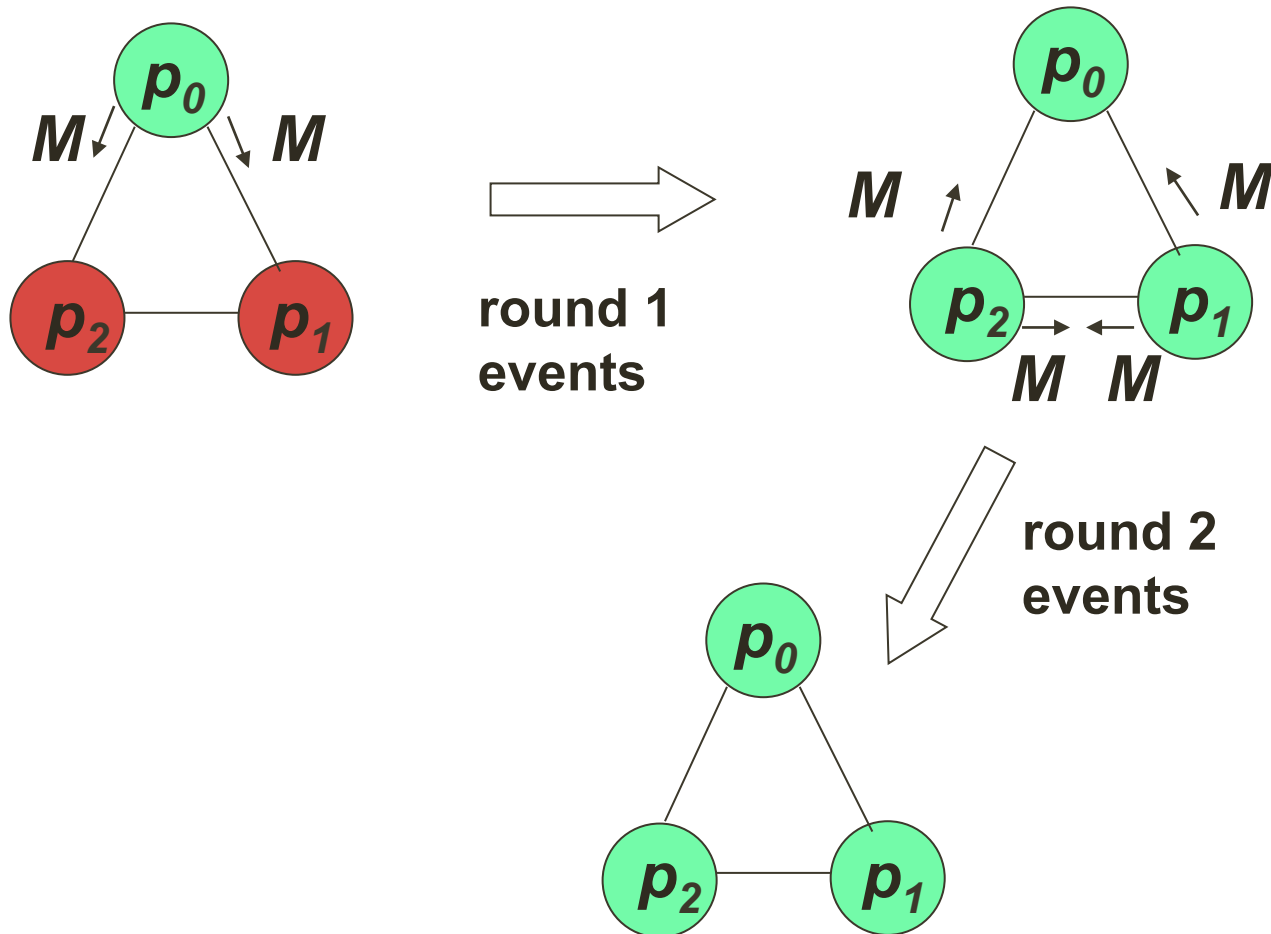
Σύγχρονες Εκτελέσεις

- Σύγχρονη Εκτέλεση: Μια **άπειρη** ακολουθία από **γύρους**
- **Γύρος**: Ακολουθία γεγονότων της μορφής **παραλαβή, ..., παραλαβή, βήμα, ..., βήμα** τέτοια ώστε:
 - **Διανέμονται όλα τα μηνύματα** που βρίσκονται σε εξερχόμενους καταχωρητές
 - **Ο κάθε επεξεργαστής εκτελεί ένα υπολογιστικό βήμα**
- **Χρονική Πολυπλοκότητα** Σύγχρονου Μοντέλου:
 - **Αριθμός των γύρων** μέχρι τον τερματισμό

Σύγχρονη Υπερχείλιση Μηνυμάτων

- Τι θα συμβεί αν ο αλγόριθμος υπερχείλισης εκτελεστεί στο Σύγχρονο Μοντέλο;
- Γύρος 1:
 - Παραλαβή του M στον p_1 από p_0
 - Παραλαβή του M στον p_2 από p_0
 - Ο p_0 δεν κάνει τίποτα
 - Ο p_1 γίνεται **πράσινος** και στέλνει το M στους p_0 και p_2
 - Ο p_2 γίνεται **πράσινος** και στέλνει το M στους p_0 και p_1
- Γύρος 2:
 - Παραλαβή του M στον p_0 από p_1 και p_2
 - Παραλαβή του M στον p_1 από p_2
 - Παραλαβή του M στον p_2 από p_1
 - Οι p_0, p_1, p_2 δεν κάνουν τίποτα

Σύγχρονη Υπερχείλιση Μηνυμάτων



Πολυπλοκότητα Αλγορίθμου

- Τερματική Κατάσταση
 - Το χρώμα του επεξεργαστή είναι πράσινο
- Πολυπλοκότητα Μηνυμάτων
 - $2m$, m είναι ο αριθμός των ακμών
- Χρονική Πολυπλοκότητα (Γύροι)
 - $\text{diam}(G)+1$, όπου $\text{diam}(G)$ η διάμετρος του γράφου G

Προσοχή: Η Πολυπλοκότητα του αλγορίθμου είναι η ίδια με αυτή στο ασύγχρονο μοντέλο. **Δεν συμβαίνει με όλους τους αλγόριθμους αυτό.**

Ερωτήσεις;

