

Προστασία Διεργασιών, User/kernel mode,
Προνομιούχες Εντολές,
Διακοπές - Εξαιρέσεις - Page Faults

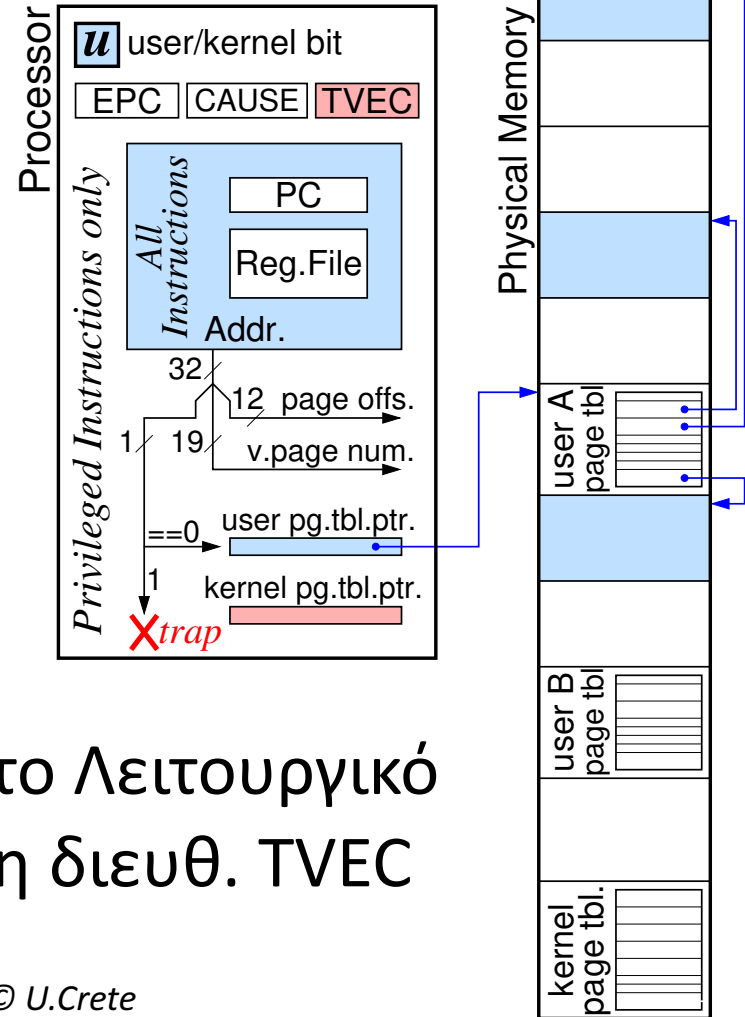
12b (§12.3) – 26 Απριλίου 2021 – Μανόλης Κατεβαίνης

Εκτέλεση Διεργασίας A, σε User Mode

Σε *User mode* απαγορεύονται:

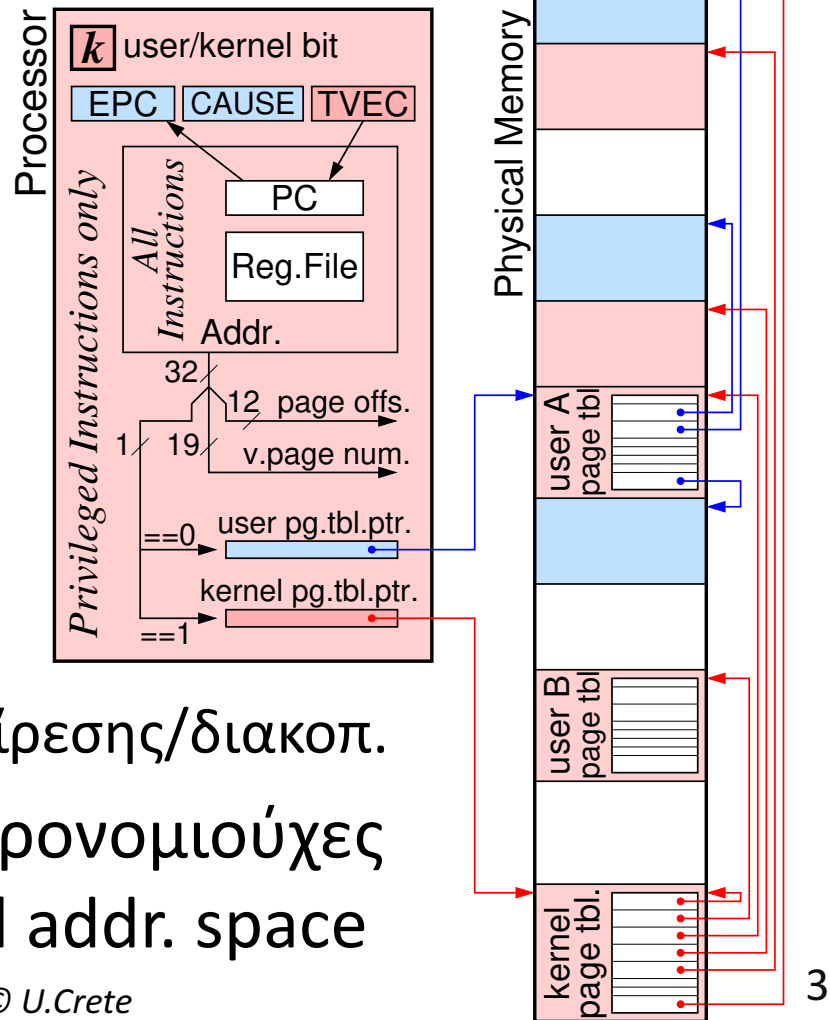
- Εκτέλεση «Προνομιούχων» εντολών (privileged instruct.)
- Προσπέλαση με MS bit εικονικής διεύθυνσης == 1
- Απόπειρα παραβίασης, ή page fault, ή ecall επιφέρει:

⇒ Exception/trap: είσοδος στο Λειτουργικό (kernel/supervisor mode) στη διευθ. TVEC



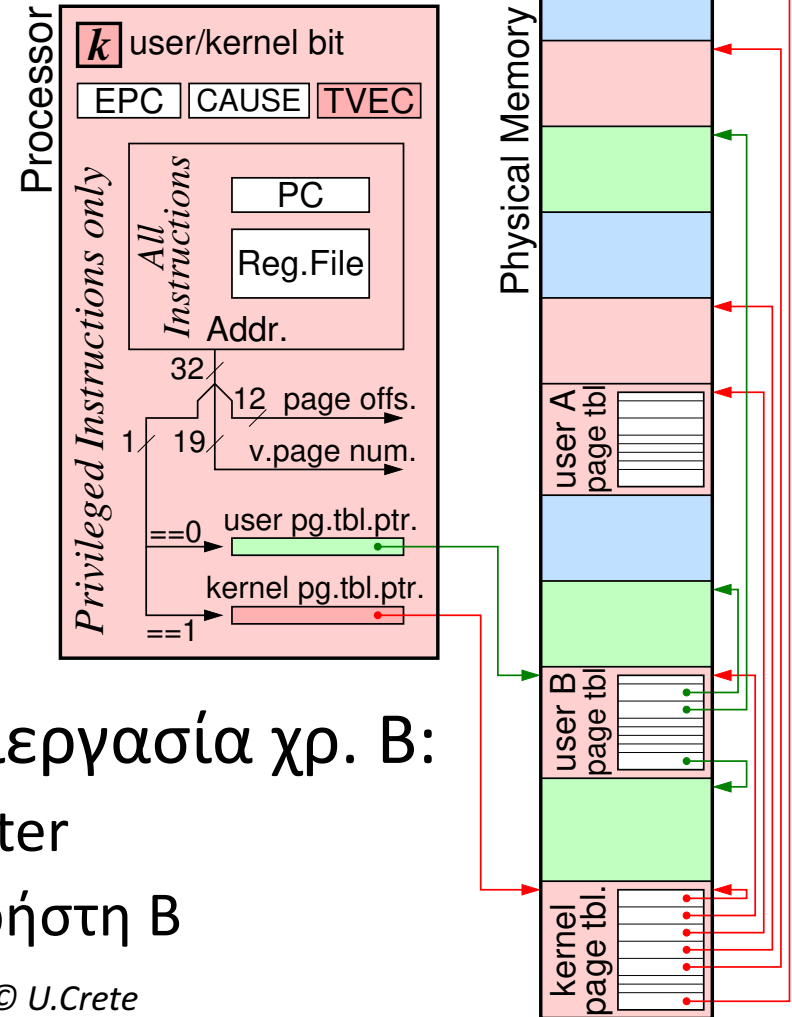
Είσοδος σε Kernel (Supervisor) Mode

- Αποθήκευση του PC της διακοπείσας εντολής στον καταχ. EPC (exception PC)
- νέος PC = TVEC (trap vector)
 - τον έχει ορίσει το Λειτουργικό
 - απαγορεύεται είσοδος αλλού!
- *set Kernel mode* (& save prev. mode)
- CAUSE reg. $\leftarrow$ κωδικός αιτίας εξαίρεσης/διακοπ.
- Σε *kernel mode* επιτρέπονται προνομιούχες εντολές και προσπελ. σε kernel addr. space



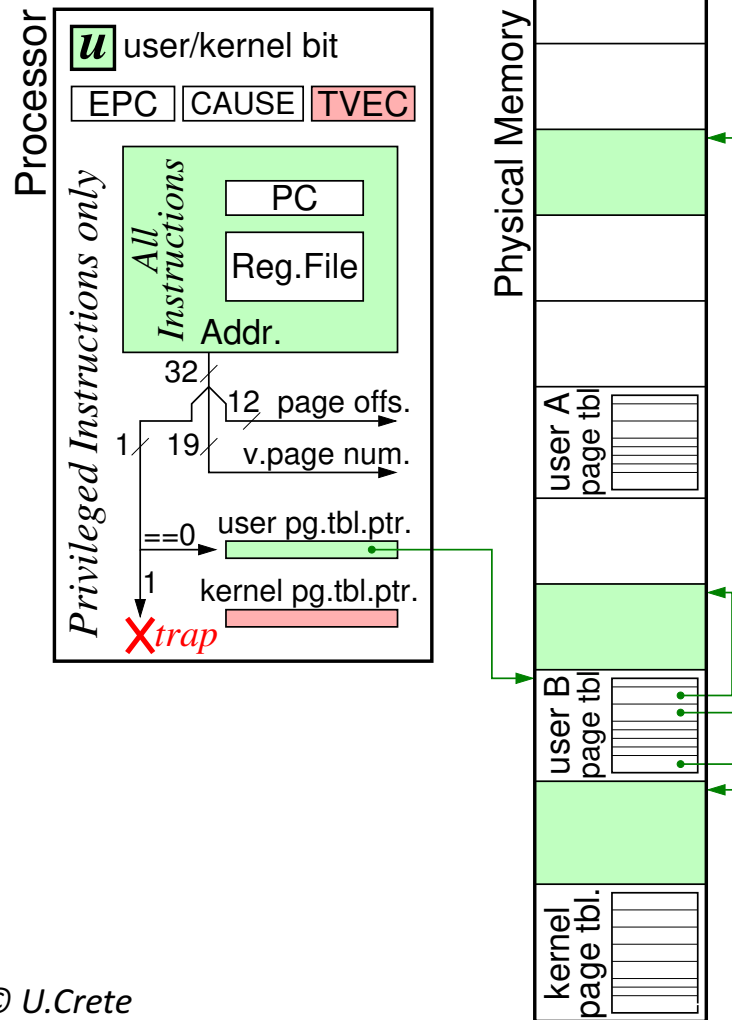
Προετοιμασία αλλαγής σε Διεργασία B

- Όταν τρέχει το Λειτουργικό (σε kernel mode, με MS addr. bit == 1), βλέπει και τον χώρο διευθύνσεων του προηγούμενου ή επόμενου χρήστη, προκειμένου να παίρνει ή δίνει ορίσματα, δεδομένα, κλπ.
- Στο σχήμα, το Λειτουργικό ετοιμάζεται να αλλάξει στην διεργασία χρ. B:
 - αλλαγή του User Page Table Pointer
 - επαναφορά καταχωρητών του χρήστη B



Επιστροφή σε Διεργασία B (User Mode)

- Set User mode
- Επαναφορά του PC σε ό,τι είχε σωθεί στον EPC όταν είχε διακοπεί η διεργασία B
- Μόνον οι σελίδες που περιέχονται στον page table της διεργασίας B είναι τώρα ορατές από τον επεξεργαστή



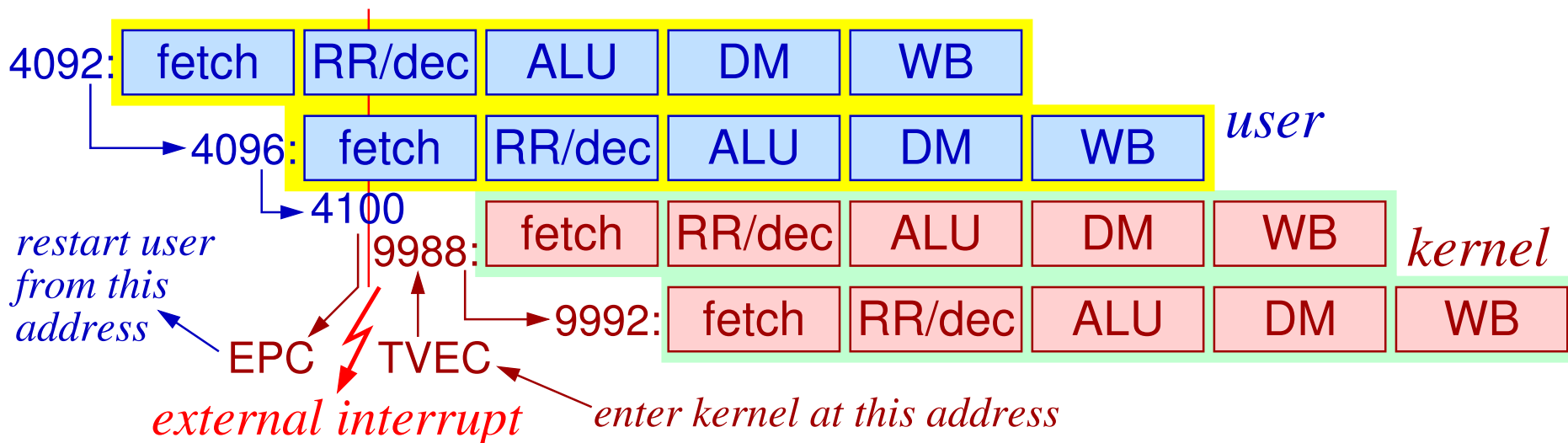
Διακοπές (Interrupts) και Εξαιρέσεις (Exceptions)

- *Interrupt*: γεγονός υψηλότερης προτεραιότητας (π.χ. περιφερειακή συσκευή) έχει ανάγκη να χρησιμοποιήσει τον επεξεργαστή, και γι' αυτό διακόπτει την διεργασία χαμηλότερης προτεραιότητας που τρέχει εκεί τώρα
- *Exception*: κάτι πήγε στραβά στην τρέχουσα διεργασία, και η εκτέλεσή της δεν μπορεί να συνεχιστεί ως έχει, αλλά πρέπει να διακοπεί, είτε τελεσίδικα, είτε προσωρινά μέχρι διόρθωση και επανεκκίνηση
- Παρόμοιος μηχανισμός και για τις δύο περιπτώσεις, ανεξαρτήτως αιτίας

Διεύθυνση Εισόδου, Vectored Interrupts

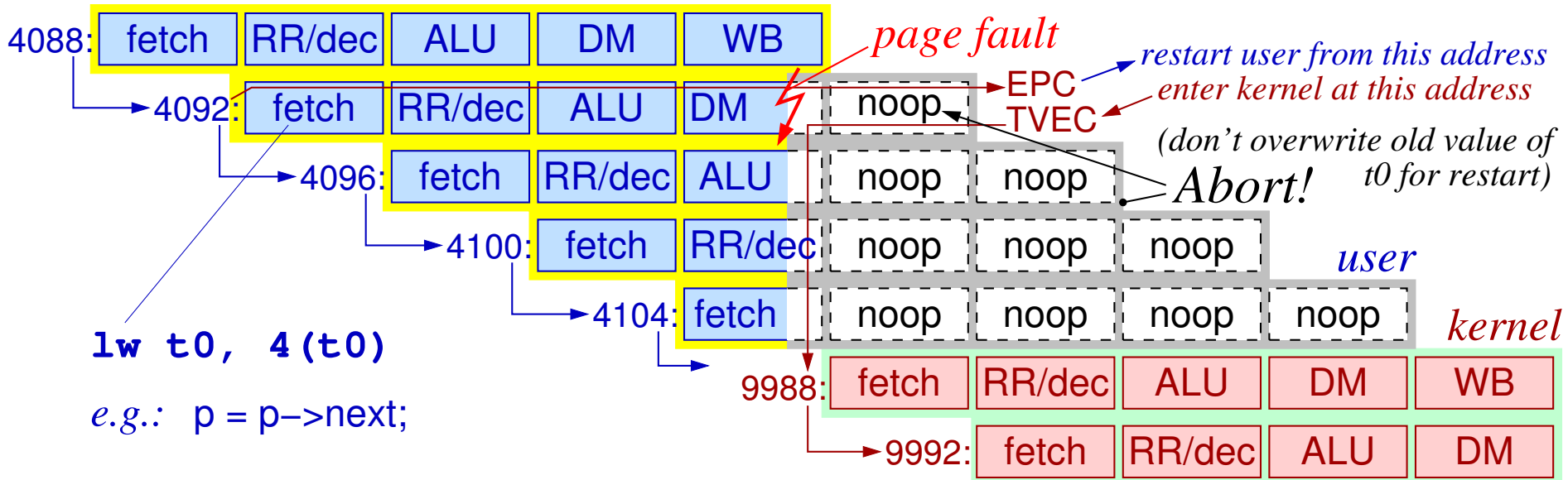
- *TVEC* (Trap Vector (ή “Supervision Trap Vector” – STVEC)): Διεύθυνση εισόδου στο Λειτουργικό, όχι υπό τον έλεγχο του χρήστη (ώστε να μη μπορεί αυτός να παρακάμπτει τους ελέγχους ασφαλείας), π.χ. από ειδικό καταχωρητή που το Λειτουργικό Σύσ. θέτει από πριν
- *CAUSE* Register (ή “Supervisor Exception Cause” – SCAUSE): κωδικοποιημένη πληροφορία για το ποιά η αιτία της διακοπής ή εξαίρεσης
- *Vectored Interrupts* – εναλλακτικός τρόπος συνδυασμού των δύο: διαφορετική διεύθυνση εισόδου για την κάθε διαφορετική αιτία
 - κάτι σαν Switch Statement βάσει αιτίας για είσοδο στο Λειτουργικό
 - πλεονέκτημα: γλιτώνουμε να το κάνουμε σε software
 - μειονέκτημα: κοινός κώδικας στην αρχή, σε όλες τις περιπτώσεις, για σώσιμο καταχωρητών κλπ. προετοιμασίες, κι ύστερα το Switch Stmt. βάσει αιτίας

Διακοπή (Interrupt) από εξωτερική αιτία (I/O etc.)



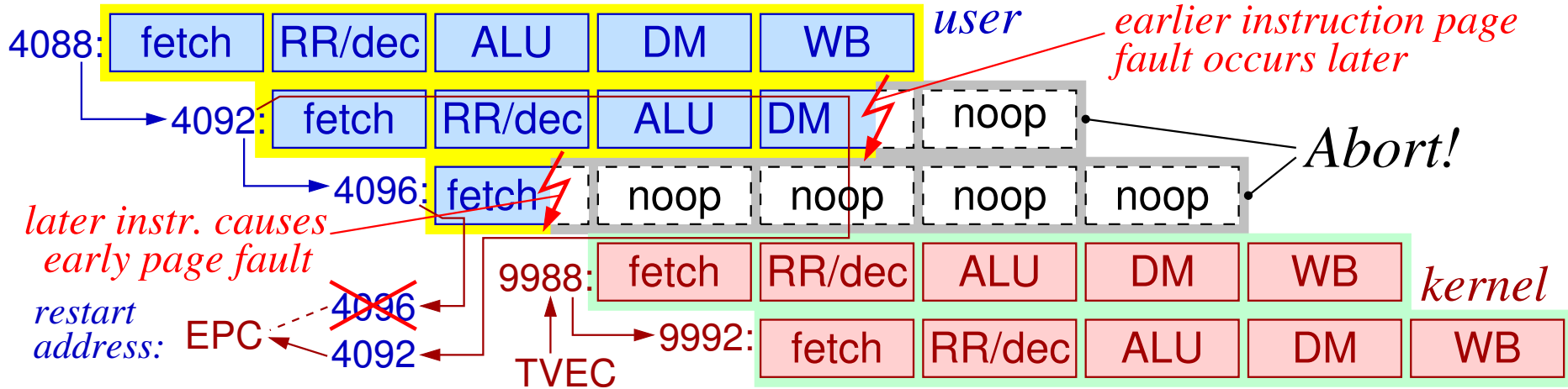
- *EPC (Exception PC)*: επόμενη εκκρεμής μετά τις εκτελεσθείσες εντολ.
- *TVEC (Trap Vector)*: η «επίσημη» είσοδος του interrupt handler
- *Προσοχή*: εάν μία από τις τελευταίες εντολές χρήστη (user) ήταν mispredicted branch ή ακυρώθηκε λόγω αναμονής εξάρτησης, τότε η τιμή που σώζεται στον EPC μπορεί να χρειαστεί διόρθωση

Page Fault: ακύρωση εντολής, με δυνατότ. επανέναρξης



- Η εντολή που προκαλεί σφάλμα σελίδας πρέπει να ακυρωθεί ως να μην είχε ξεκινήσει ποτέ, και να επανεκτελεστεί μετά την εξυπηρ. του page fault
 - OK σε RISC (μόνο μία εγγρ. ανά εντολή), δυσκολότερο σε CISC (πολλαπλές εγγραφές)
- Ομοίως και οι μετά από αυτήν, για «καθαρή» επανεκκίνηση του προγραμ.

Προσοχή στις πολλαπλές εξαιρέσεις εντός Pipeline



- Πρώτα διαπιστώνεται το page fault επόμενης εντολής
- Μετά διαπιστώνεται και δεύτερο page fault νωρίτερης εντολής
- Τελικά πρέπει να καταγραφεί η νωρίτερη εντολή σαν αιτία, διότι από την νωρίτερη πρέπει να ξαναρχίσει η εκτέλεση μετά την εξυπηρέτηση του (ή καλύτερα των) σφάλματος/σφαλμάτων σελίδας