

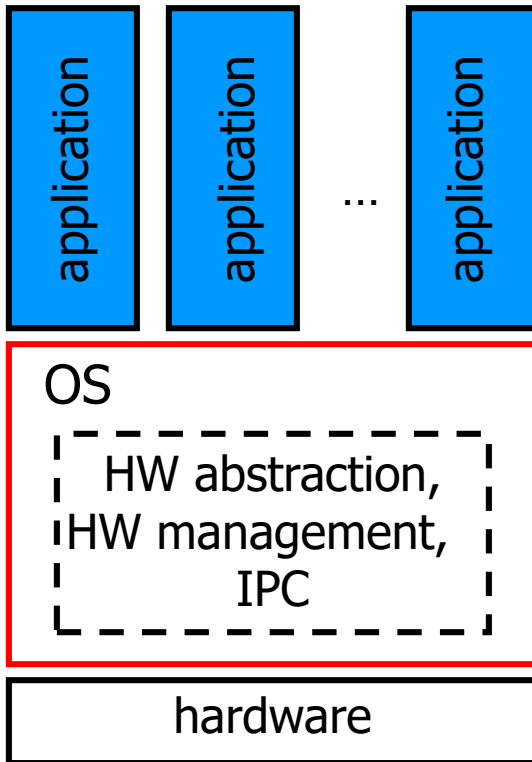
Το λειτουργικό σύστημα

Συστήματα υπολογιστών

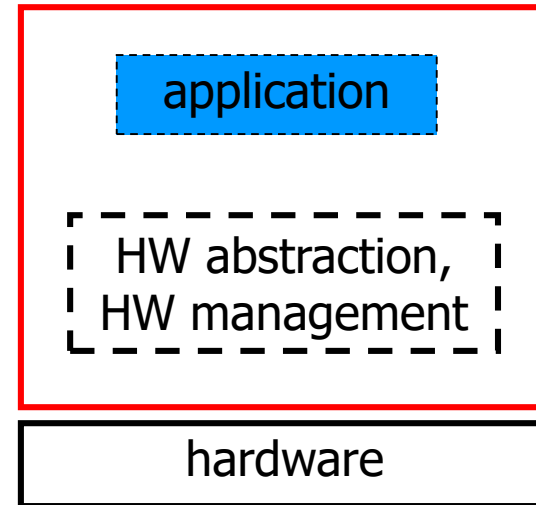
- Ειδικού σκοπού
 - συστήματα για μια συγκεκριμένη εφαρμογή
 - η εφαρμογή είναι γνωστή εκ των προτέρων
 - περιορισμένοι υπολογιστικοί / αποθηκευτικοί πόροι
 - δεν τίθεται θέμα υποστήριξης πολλών χρηστών
- Γενικής χρήσης
 - υπάρχουν πολλές (ανεξάρτητες) εφαρμογές που μπορεί να εκτελούνται ταυτόχρονα μεταξύ τους
 - μπορεί να δίνεται πρόσβαση σε πολλούς χρήστες που (γενικά) δεν εμπιστεύονται ο ένας τον άλλο
 - ο αριθμός και το είδος των εφαρμογών είναι ανοιχτός και στην γενική περίπτωση μπορεί να αλλάξει δυναμικά



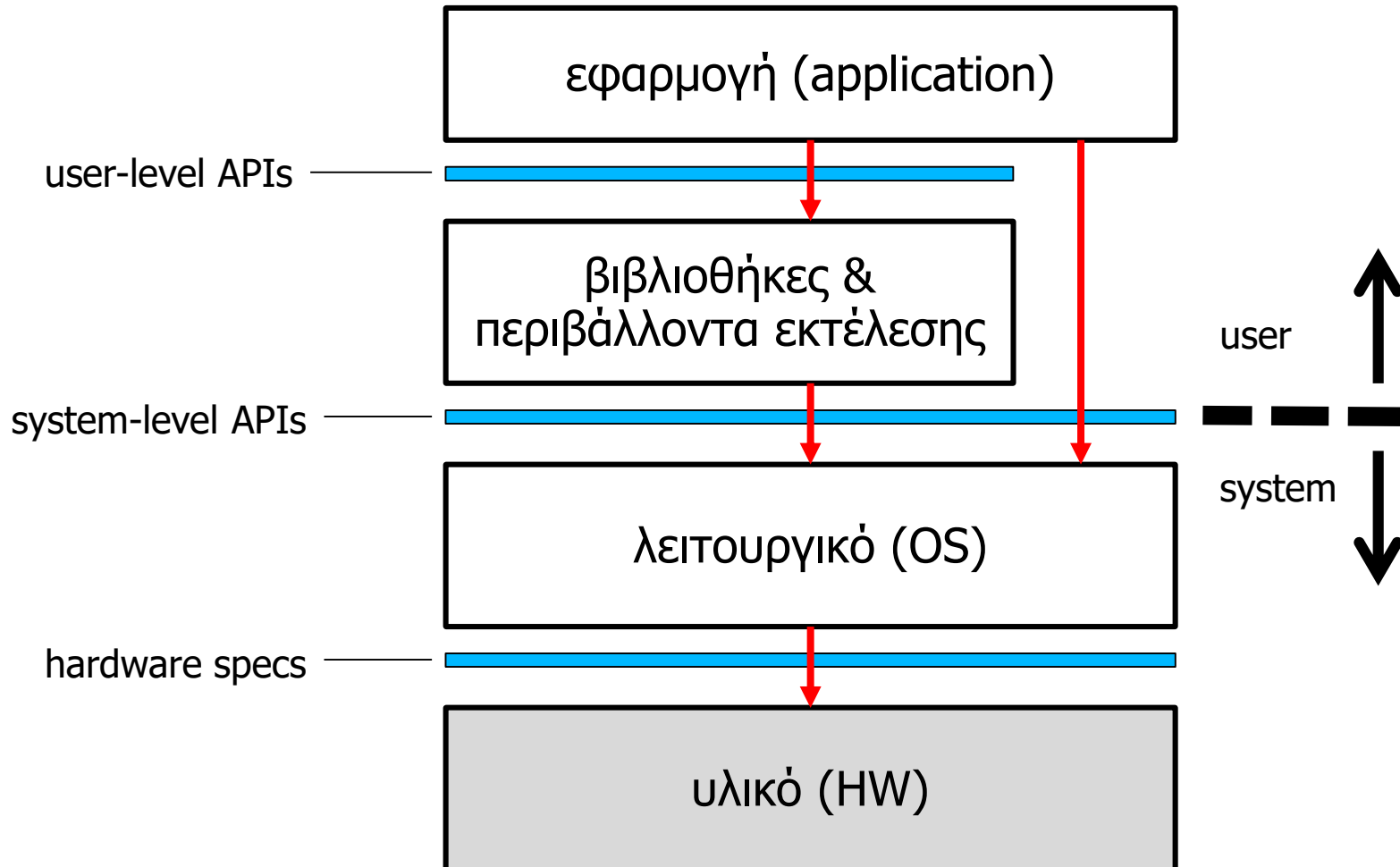
σύστημα γενικής χρήσης



σύστημα ειδικού σκοπού



Ενδεικτική στοίβα λογισμικού ενός Η/Υ



Λειτουργικό σύστημα και πυρήνας

- **Λειτουργικό σύστημα (operating system)**
το σύνολο του λογισμικού και των διεργασιών μέσω των οποίων υλοποιείται η συνολική λειτουργικότητα που παρέχεται στις εφαρμογές
- **Πυρήνας λειτουργικού (OS kernel)**
το τμήμα του λειτουργικού συστήματος που υλοποιεί την επικοινωνία και διαχείριση του υλικού στο πιο χαμηλό επίπεδο
- Εδώ δεν μας ενδιαφέρει αυτός ο διαχωρισμός ...

Βασικοί ρόλοι και μηχανισμοί ΛΣ

- Αφαίρεση (abstraction) του υλικού
 - απόκρυψη λεπτομερειών υλοποίησης, ανεξαρτησία από υλικό
- Διαχείριση πόρων και συσκευών του συστήματος
 - επεξεργαστής, μνήμη, δίσκος, δίκτυο, λοιπά περιφερειακά
- Υποστήριξη εκτέλεσης προγραμμάτων εφαρμογής
- Υποστήριξη αποθήκευσης δεδομένων
- Υποστήριξη επικοινωνίας προγραμμάτων
- Κατανομή και ελεγχόμενη πρόσβαση σε πόρους του συστήματος

Ασφάλεια

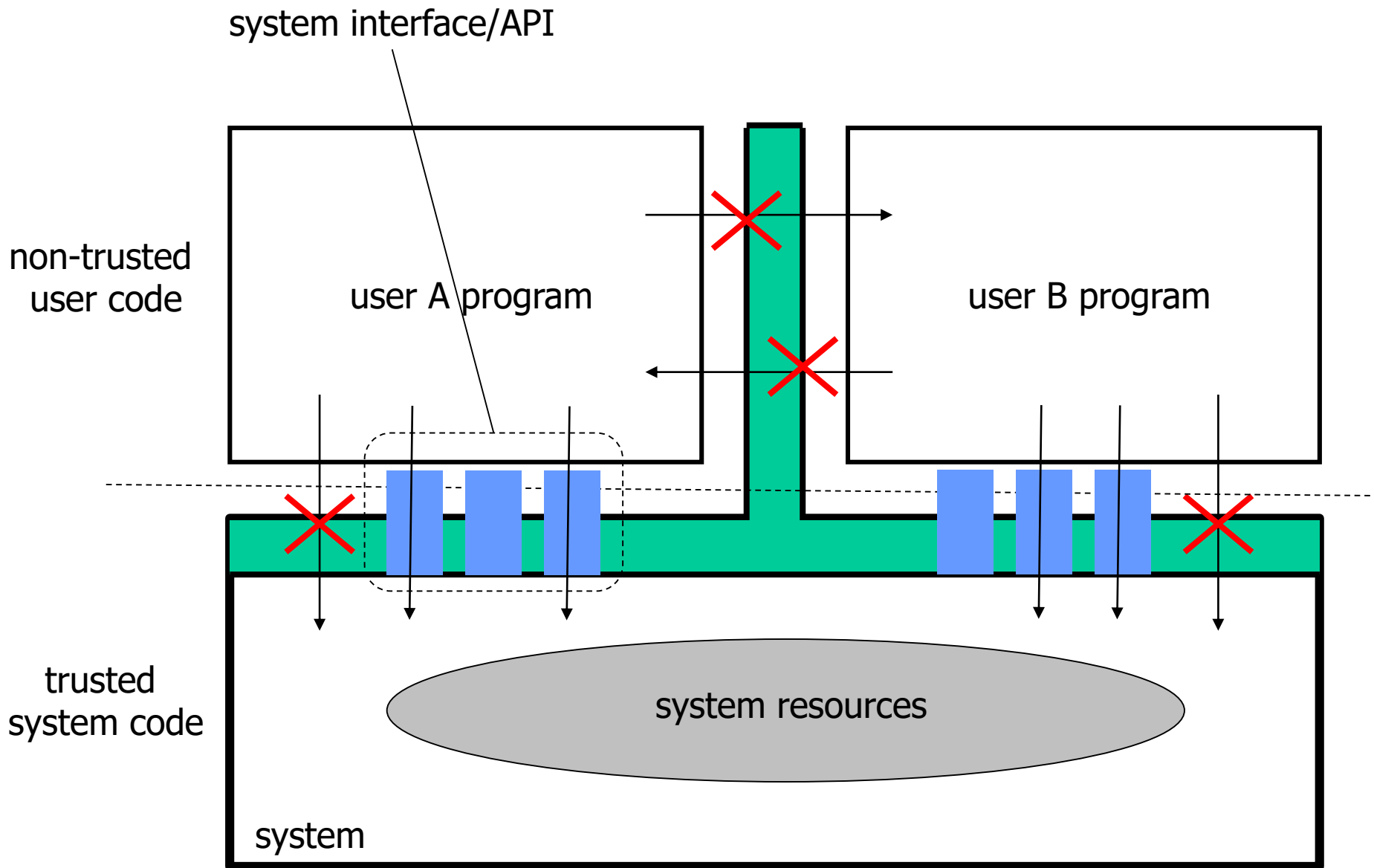
- Δεν μπορεί να γίνει υπόθεση για την ορθή συμπεριφορά προγραμμάτων και χρηστών
 - προγραμματιστικά λάθη, κακόβουλοι χρήστες/προγράμματα
- Το λειτουργικό παρέχει κατάλληλους **μηχανισμούς ελέγχου/ασφάλειας**
- Δύο διαφορετικά επίπεδα/διαστάσεις
- **Προστασία του ΛΣ** από εφαρμογές / χρήστες
- **Προστασία εφαρμογών / χρηστών** μεταξύ τους

Διαχωρισμός σύστημα - χρήστη

- Η ασφάλεια στα λειτουργικά συστήματα βασίζεται στον **διαχωρισμό** μεταξύ του κώδικα που ανήκει στο **σύστημα** και του κώδικα που ανήκει στον **χρήστη**
- Κώδικας συστήματος
 - υλοποιεί τις βασικές λειτουργίες του συστήματος
 - γράφεται από εξειδικευμένους/έμπιστους προγραμματιστές
 - θεωρείται **αξιόπιστος** (χωρίς αυτό να ισχύει πάντα)
- Κώδικας χρήστη
 - υλοποιεί λειτουργικότητα της εφαρμογής
 - μπορεί να γραφτεί από οποιονδήποτε
 - μπορεί να έχει πολλά και διάφορα λάθη
 - μπορεί να είναι «κακόβουλος»

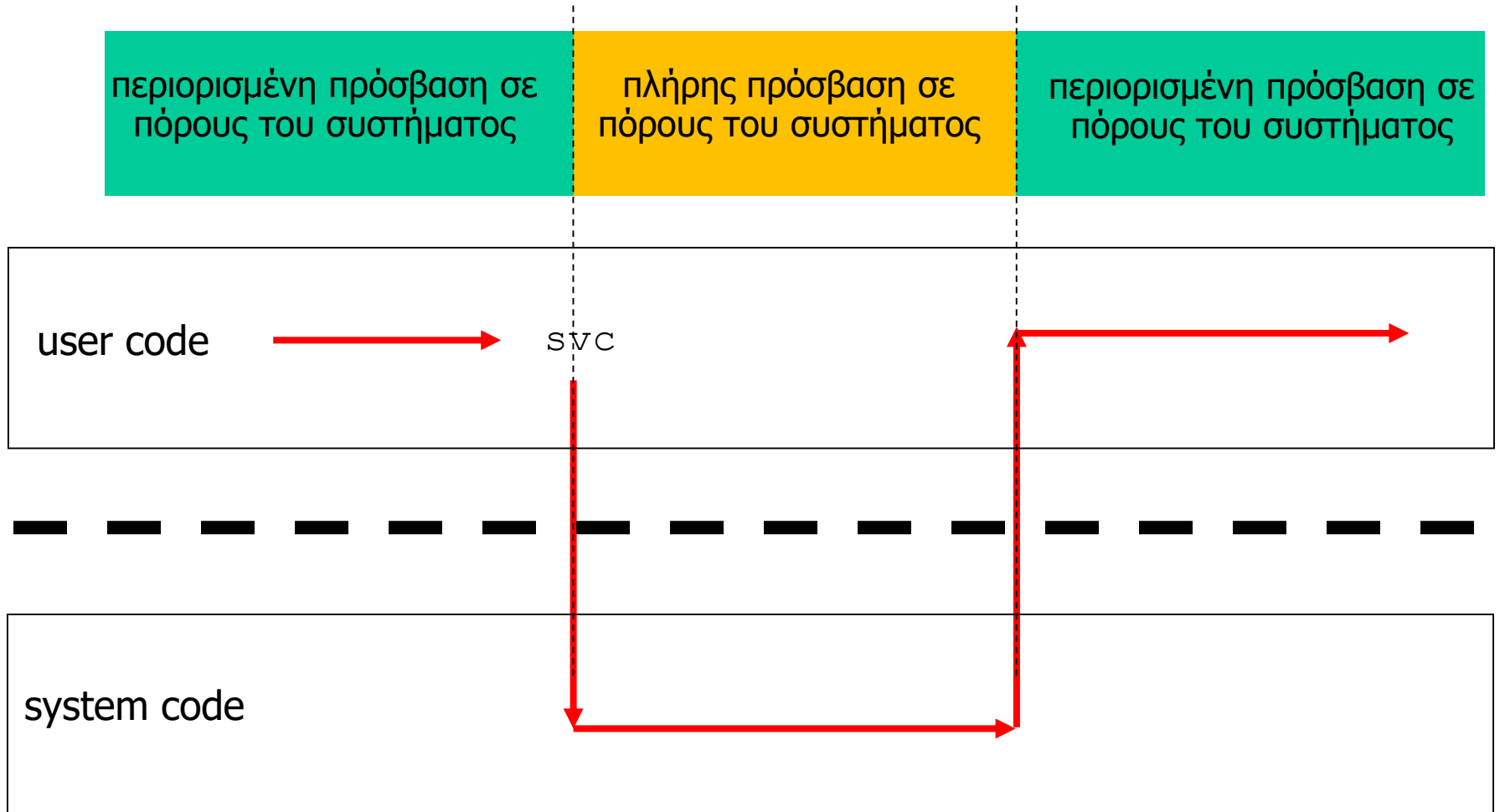
Διεπαφή λειτουργικού συστήματος

- Ο κώδικας χρήστη **δεν** έχει απ' ευθείας πρόσβαση σε κρίσιμους πόρους και συσκευές του συστήματος
- Το λειτουργικό παρέχει **συγκεκριμένες** λειτουργίες
 - επικοινωνίας του κώδικα χρήστη με το λειτουργικό
 - επικοινωνίας μεταξύ διαφορετικών προγραμμάτων χρήστη
- Η πρόσβαση σε αυτές τις λειτουργίες γίνεται μέσω ειδικών **κλήσεων συστήματος** (system calls)
- Το σύνολο των κλήσεων συστήματος ονομάζεται και **διασύνδεση συστήματος** (system interface)



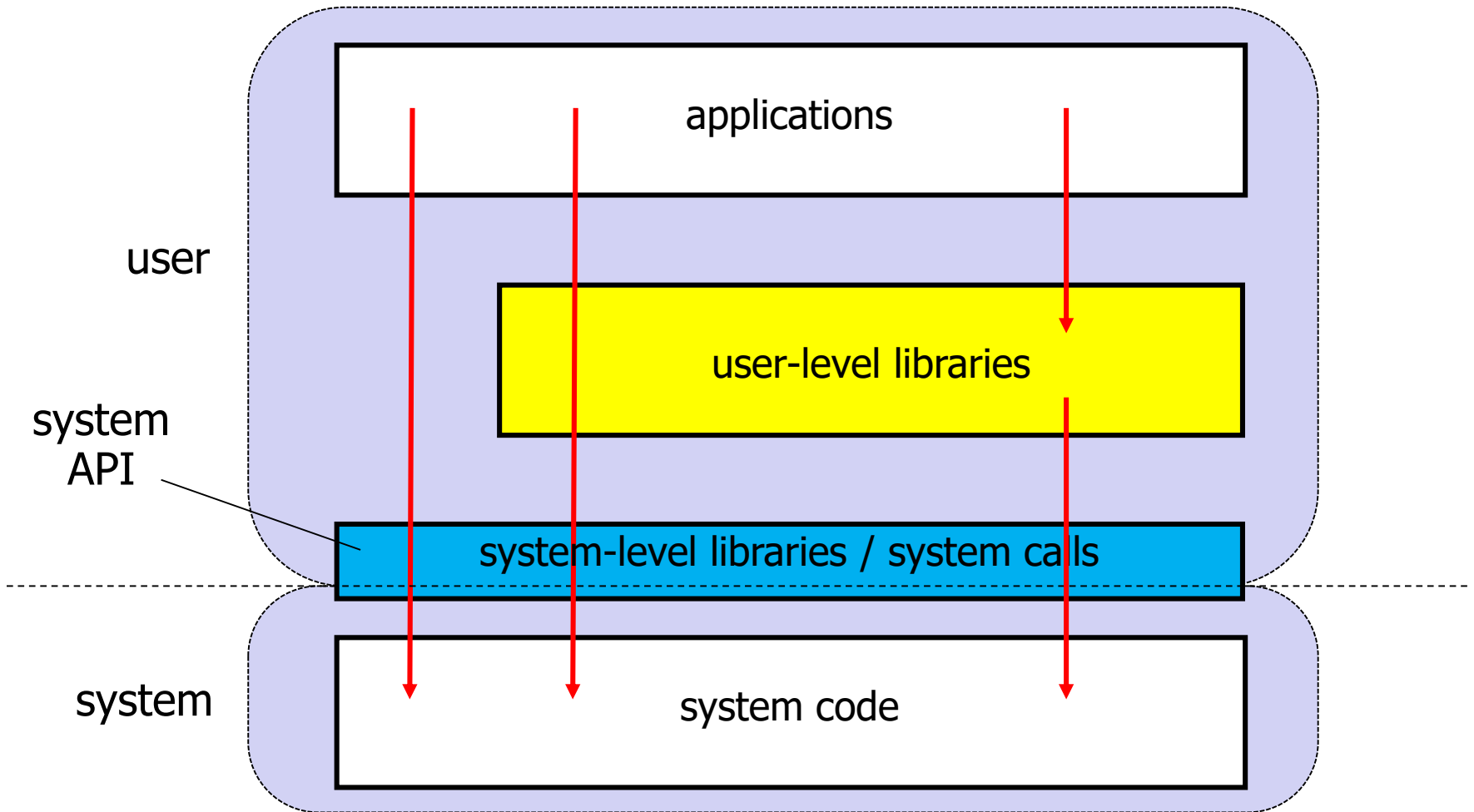
Κλήσεις συστήματος

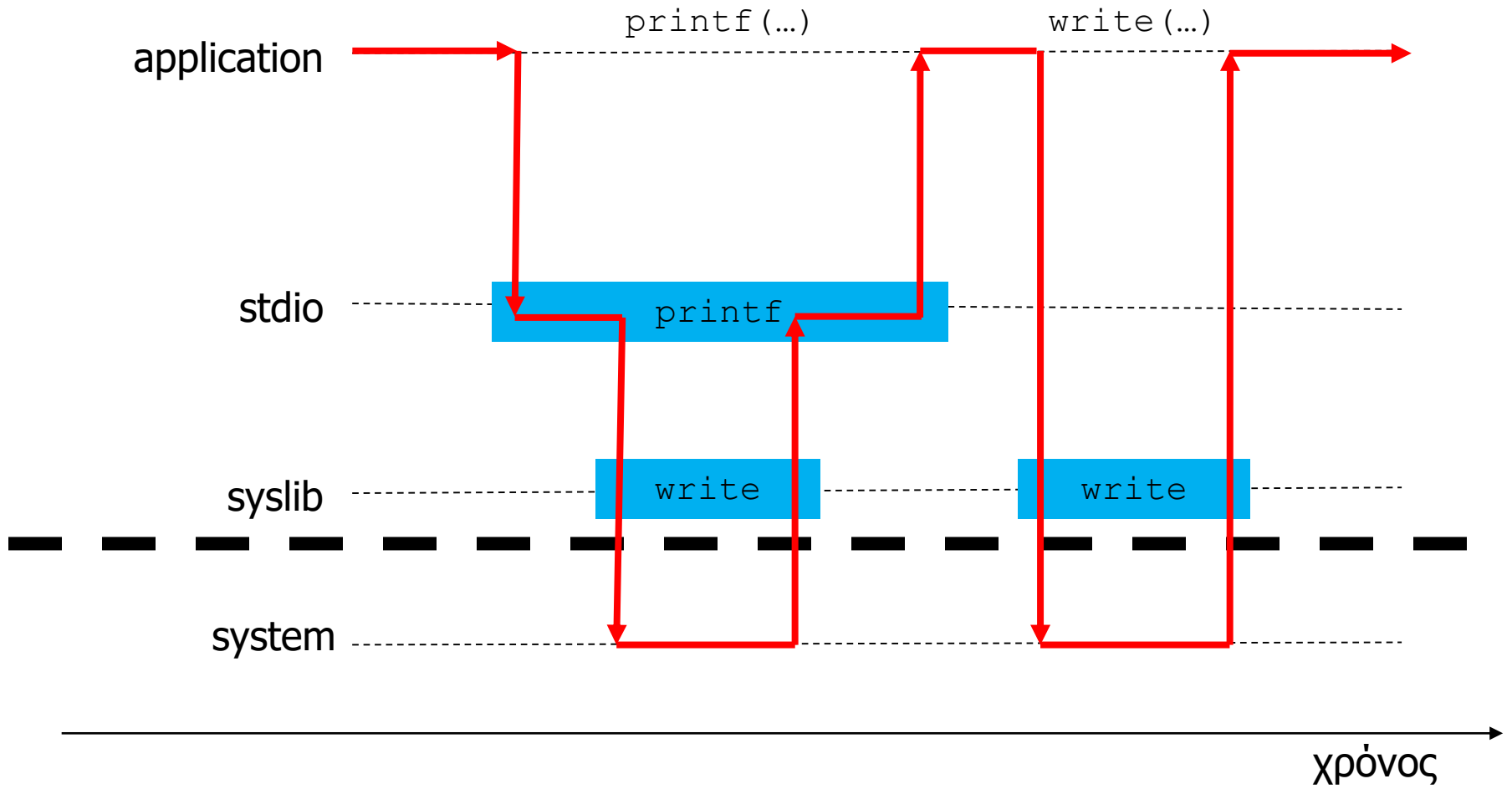
- Οι κλήσεις συστήματος είναι **ειδικές** κλήσεις συνάρτησης
 - το άλμα στον κώδικα του λειτουργικού υλοποιείται μέσω ειδικής εντολής του επεξεργαστή (supervisor call – svc)
- Όταν πραγματοποιείται μια κλήση συστήματος
 - **αλλαγή** του επεξεργαστή από user σε system mode
 - άλμα στον **κώδικα του λειτουργικού**
- Όταν τερματιστεί η κλήση συστήματος
 - επιστροφή στον **κώδικα του χρήστη**
 - **επαναφορά** του επεξεργαστή σε user mode
- Η πρόσβαση στις εσωτερικές λειτουργίες και τους πόρους του ΛΣ επιτρέπεται **μόνο** σε system mode
 - προστασία από bugs και «κακόβουλο» (malicious) κώδικα



Απόκρυψη κλήσεων συστήματος

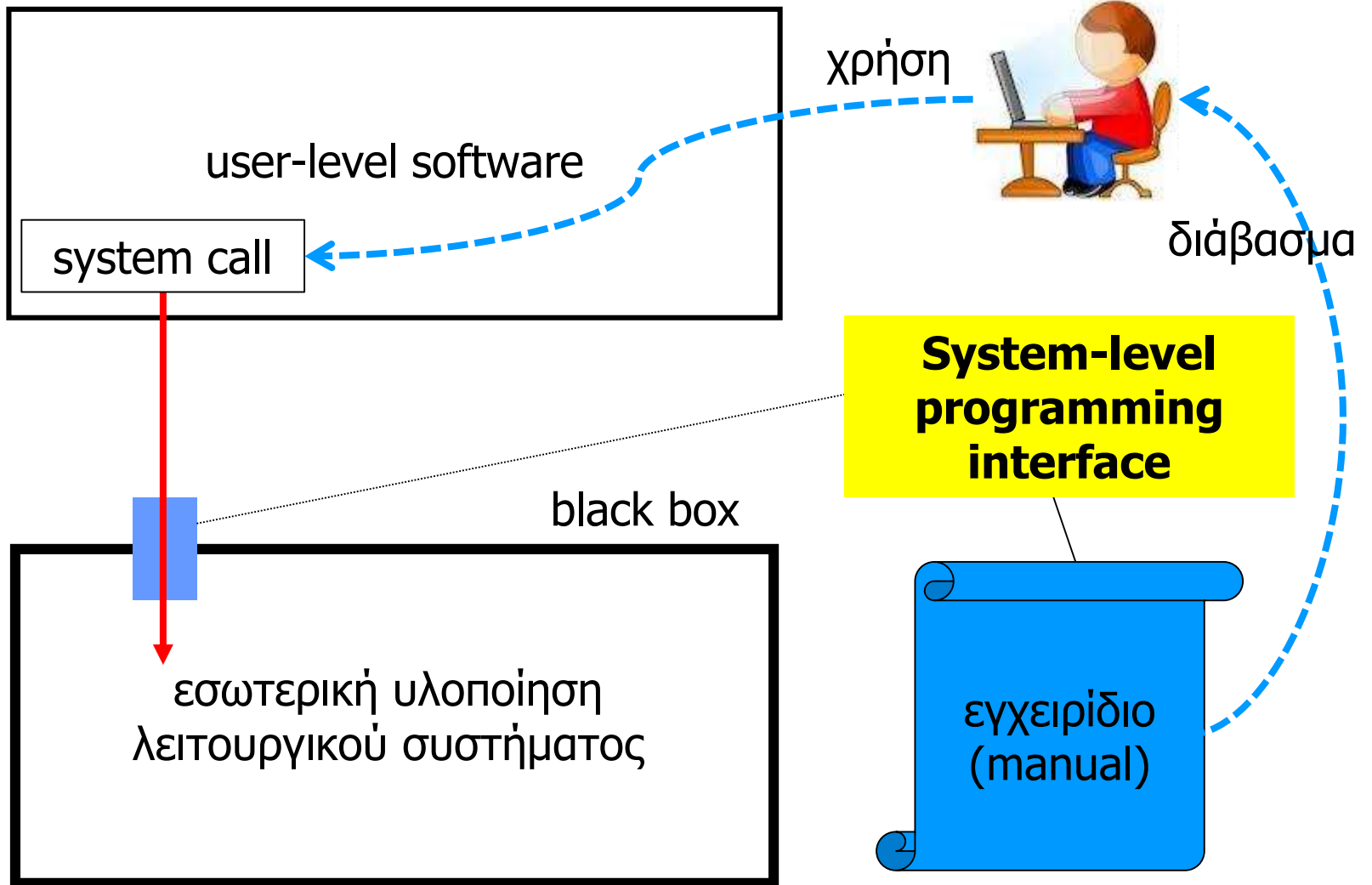
- Οι συμβάσεις κλήσης είναι περίπλοκες
 - γίνονται σε χαμηλό επίπεδο
 - αποθήκευση παραμέτρων και επιστροφή αποτελεσμάτων
 - εύκολα γίνονται προγραμματιστικά λάθη
- Οι κλήσεις συστήματος συνήθως «περιτυλίγονται» ώστε να έχουν την μορφή οικείων συναρτήσεων, π.χ., με βάση την σύνταξη/συμβάσεις της C
- Τα περιτυλίγματα ορίζονται σε ειδικές βιβλιοθήκες
 - system-level libraries

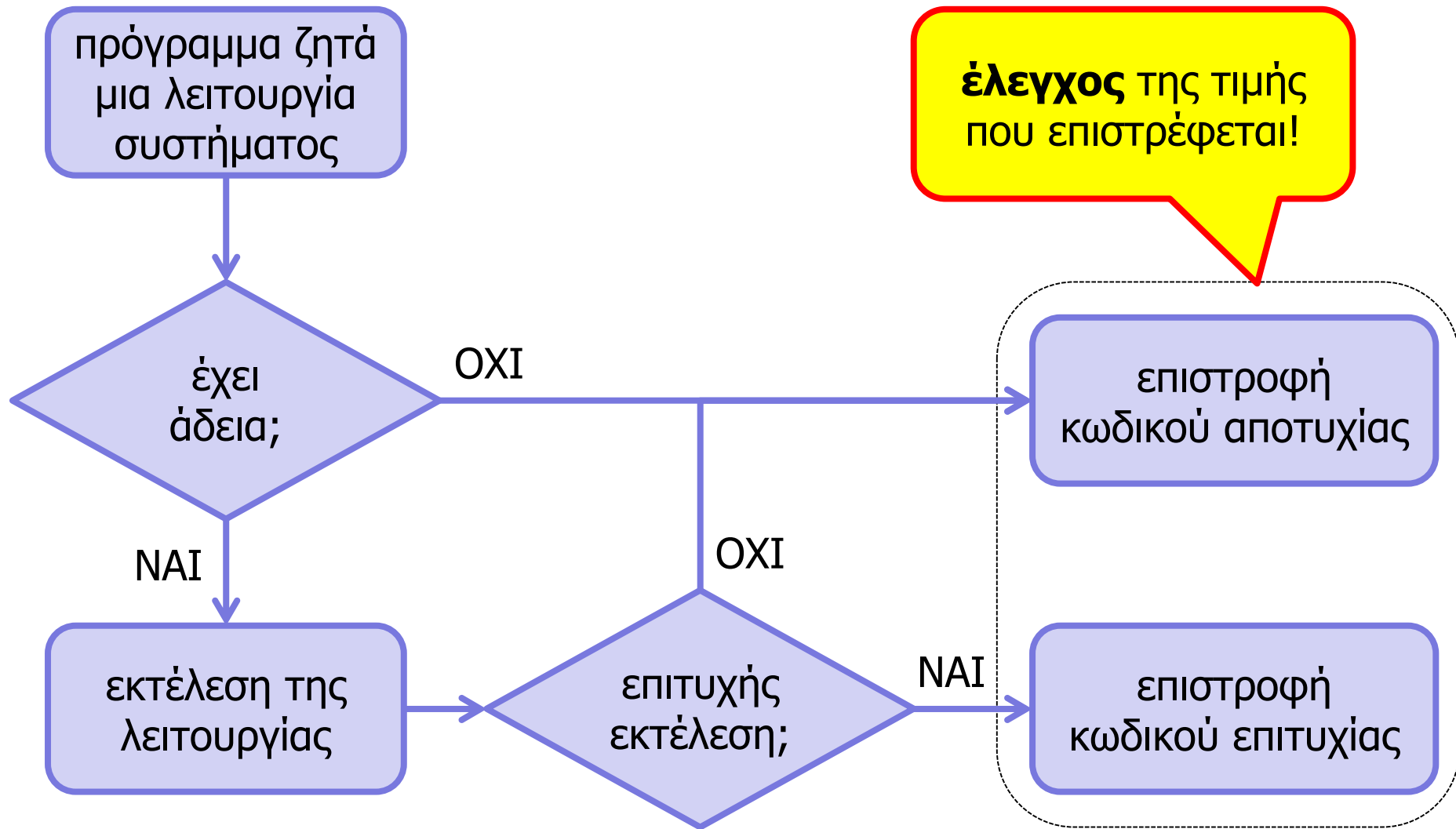




Το λειτουργικό σύστημα ως μαύρο κουτί

- Προσφέρει προγραμματιστική διασύνδεση/διεπαφή με βάση την οποία μπορεί να προγραμματιστούν πολλές διαφορετικές εφαρμογές
- Ο προγραμματιστής δεν χρειάζεται να γνωρίζει τις εσωτερικές λεπτομέρειες υλοποίησης του λειτουργικού συστήματος
- Πρέπει όμως να γνωρίζει ποιες λειτουργίες προσφέρονται από το λειτουργικό – και πώς να τις χρησιμοποιεί σωστά

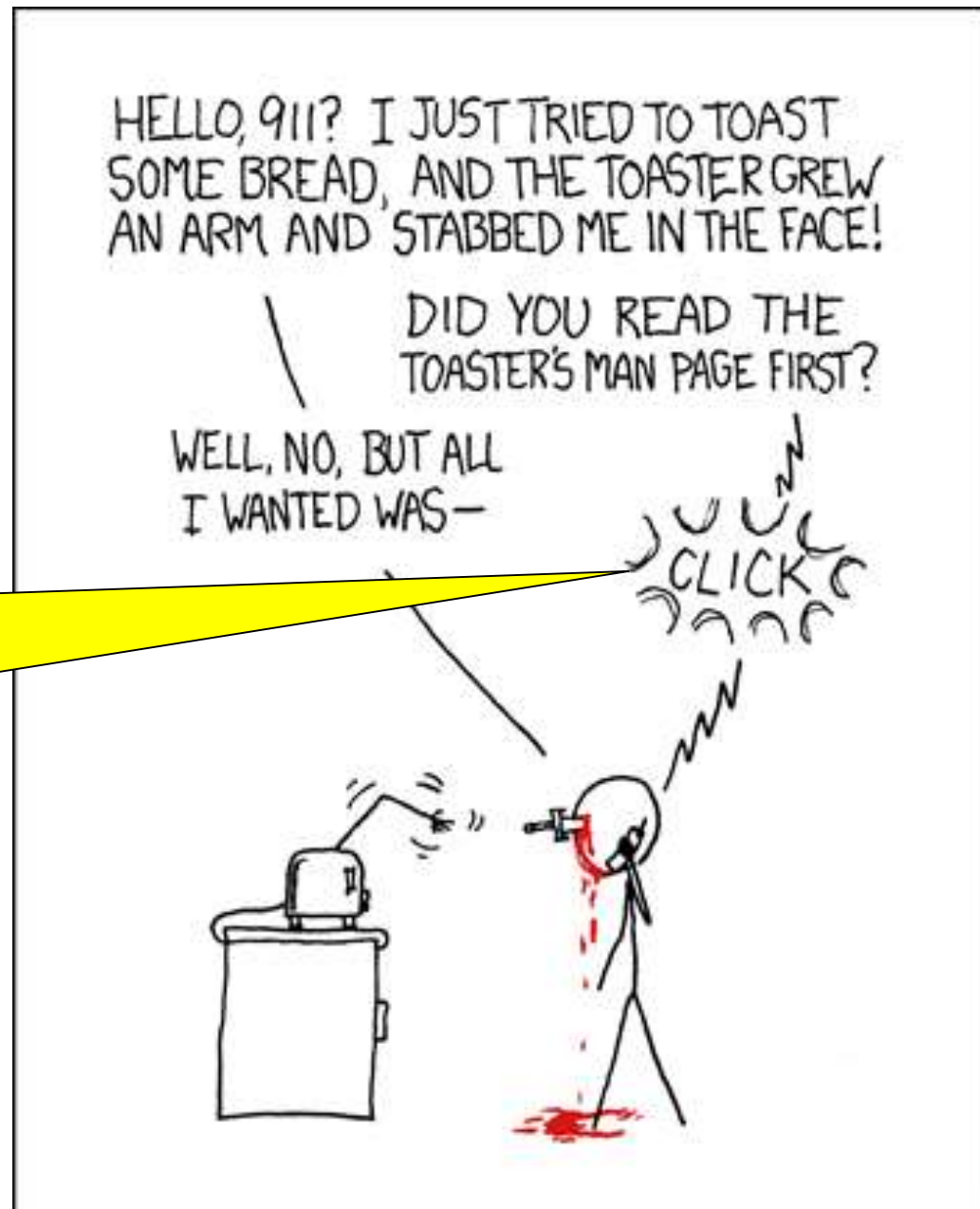




Αποτυχία κλήσης συστήματος

- Η τιμή που επιστρέφει μια κλήση συστήματος πρέπει **οπωσδήποτε(!)** να ελέγχεται
- Σηματοδοτεί αν η κλήση πέτυχε (ενίοτε μαζί με επιπλέον πληροφορία) ή αν απέτυχε (όχι τον λόγο της αποτυχίας)
- Ο **λόγος αποτυχίας** αποθηκεύεται, ως κωδικός, στην καθολική μεταβλητή συστήματος `errno`
 - οι κωδικοί λαθών ορίζονται στο αρχείο `errno.h`
 - η συνάρτηση `strerror` επιστρέφει το (αλφαριθμητικό) μήνυμα που αντιστοιχεί στην τρέχουσα τιμή της `errno`
 - η συνάρτηση `perror` εκτυπώνει αυτό το μήνυμα
 - η τιμή της `errno` ανανεώνεται **μόνο** σε περίπτωση αποτυχίας μια κλήσης συστήματος – γιατί;

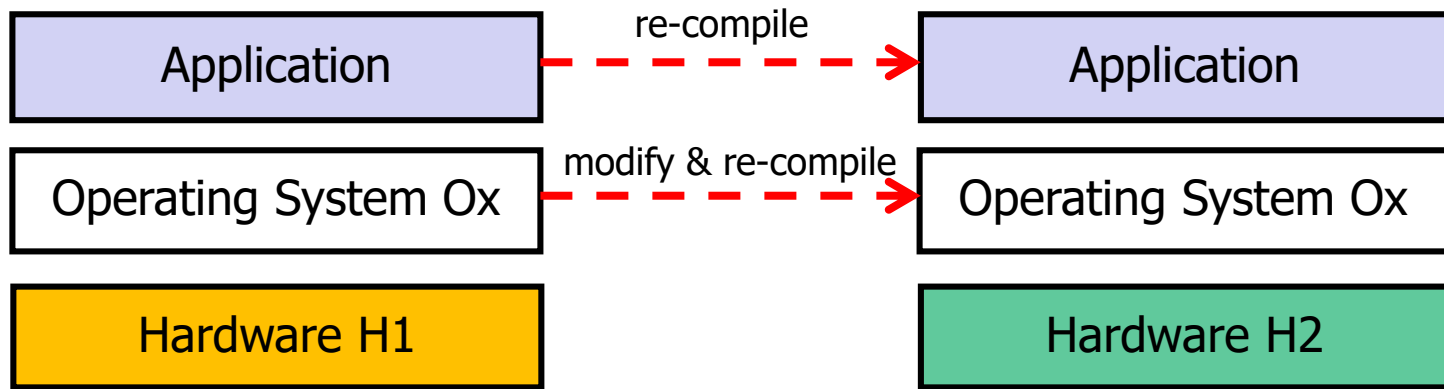
RTFM



Το πρόβλημα της φορητότητας (portability)

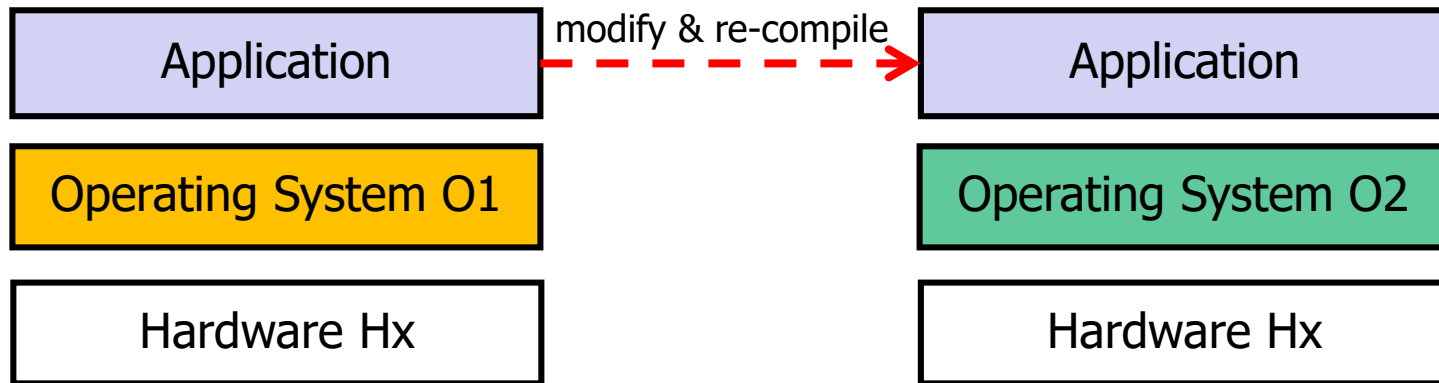
- Συχνά είναι επιθυμητό ο κώδικας που γράφουμε να μπορεί να εκτελεστεί (σωστά), **χωρίς καμία αλλαγή**, σε διαφορετικούς υπολογιστές
 - διαφορετικό υλικό/επεξεργαστές, διαφορετικά λειτουργικά
- Η φορητότητα του κώδικα μπορεί να είναι καθοριστικής σημασίας για την διάδοση του
- Όσο πιο κοντά στο λειτουργικό σύστημα βρίσκεται ο κώδικας τόσο λιγότερο μεταφέρισμος είναι
 - μπορεί όμως να εκμεταλλευτεί ειδικές λειτουργίες
- Υπάρχουν προδιαγραφές λειτουργιών επιπέδου συστήματος ανεξάρτητες λειτουργικού συστήματος
 - POSIX (Portable Operating System Interface for Unix)

Μεταφορά σε διαφορετική πλατφόρμα υλικού



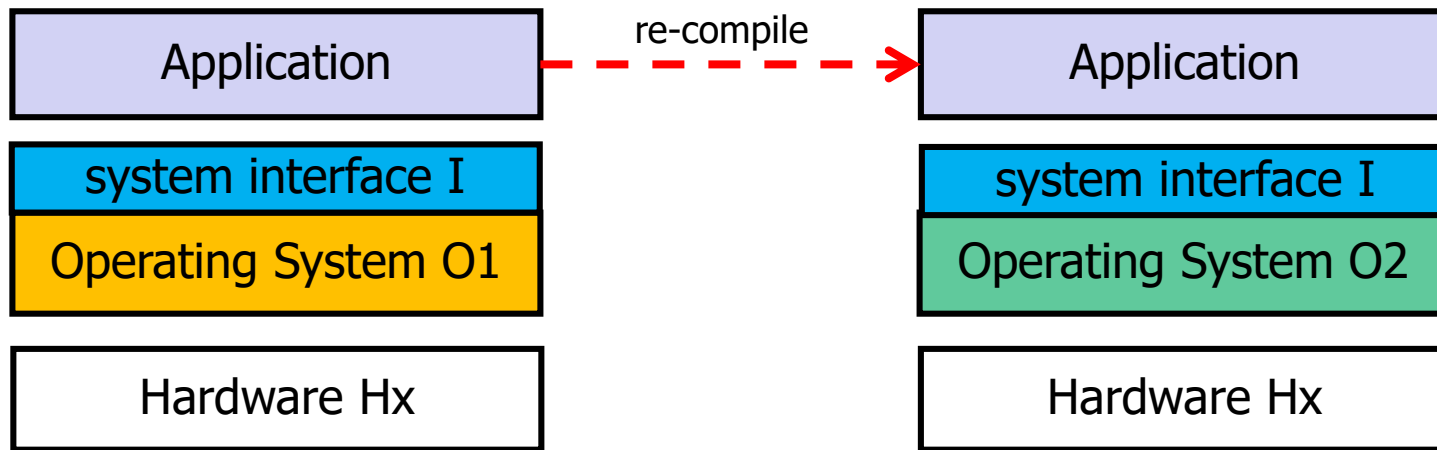
- Αλλαγή/προσαρμογή & μετάφραση λειτουργικού
- Μετάφραση προγράμματος εφαρμογής

Μεταφορά σε διαφορετικό λειτουργικό σύστημα



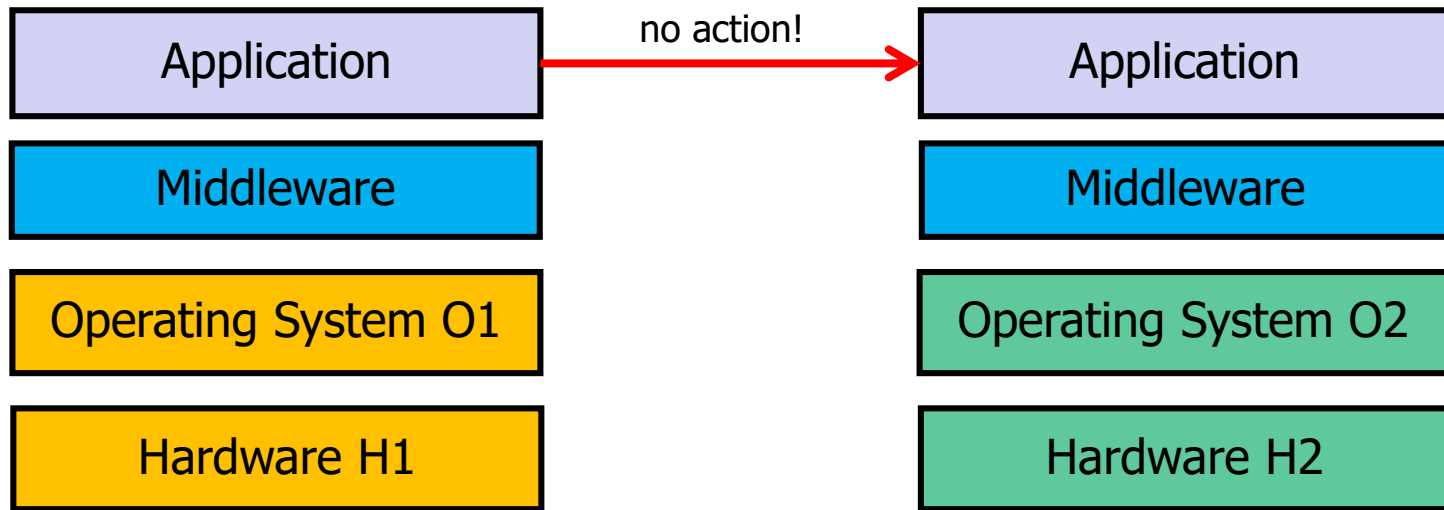
- Μετάφραση προγράμματος εφαρμογής

Μεταφορά σε διαφορετικό λειτουργικό σύστημα με ίδια διεπαφή προγραμματισμού



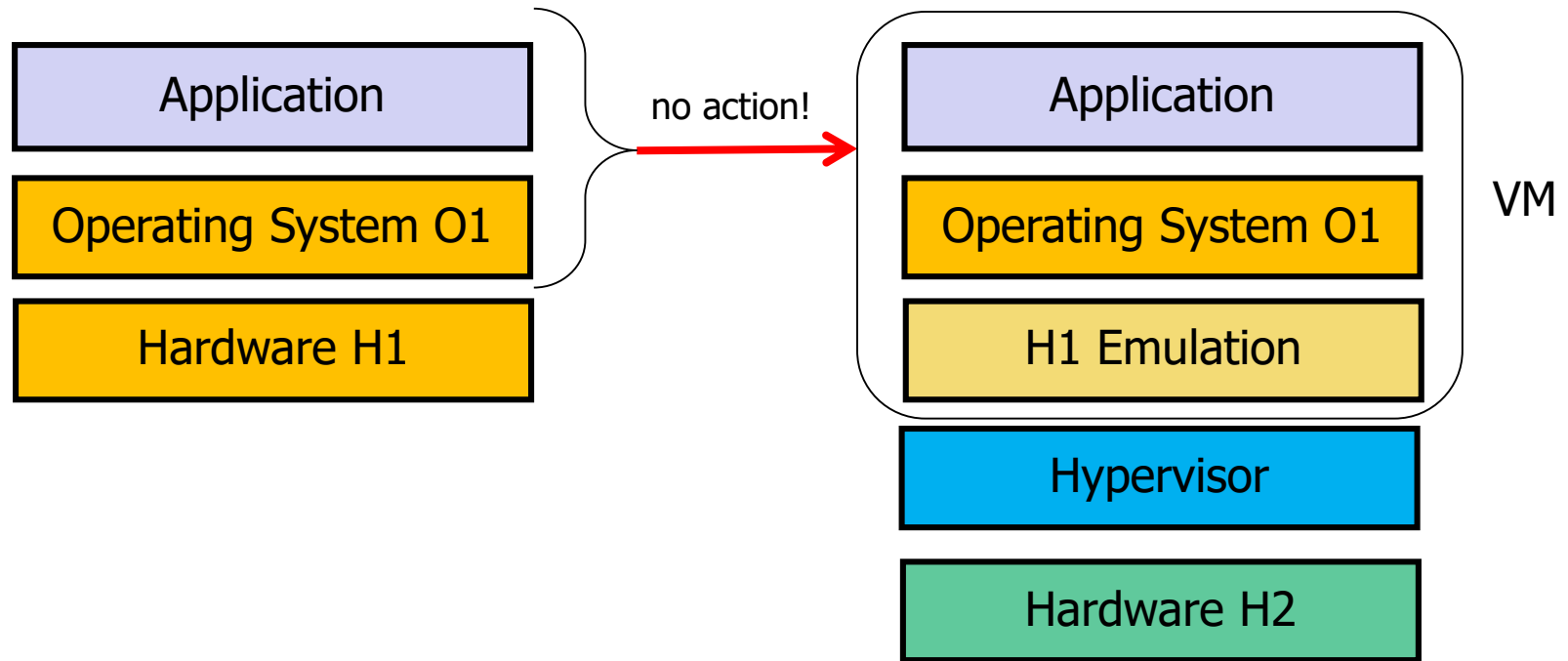
- Μετάφραση προγράμματος εφαρμογής

Μεταφορά σε διαφορετικό λειτουργικό σύστημα ή/και πλατφόρμα υλικού χωρίς παρέμβαση



- Ο προγραμματιστής δεν χρειάζεται να κάνει τίποτα
- Χρησιμοποιείται ειδικό ενδιάμεσο/μεσοστρωματικό λογισμικό (middleware) που εξασφαλίζει ομοιόμορφη εσωτερική αναπαράσταση για όλα τα δεδομένα και τις λειτουργίες της εφαρμογής και του συστήματος

Μεταφορά μέσω εικονικής μηχανής



- Ο προγραμματιστής δεν χρειάζεται να κάνει τίποτα
- Η εφαρμογή πακετάρεται **μαζί** με το λειτουργικό, και εκτελείται στο πλαίσιο μιας «εικονικής μηχανής» που εξομοιώνει το υλικό για το οποίο έχει φτιαχτεί το λειτουργικό, πάνω από άλλο λειτουργικό σύστημα