

Μηχανισμοί Προστασίας (Protection Mechanisms)

- κάθε διεργασία που εκτελείται σ' ένα λειτουργικό σύστημα πρέπει να προστατευτεί από τις ενέργειες των άλλων διεργασιών που συνυπάρχουν με αυτήν
- οι τεχνικές λύσης του προβλήματος της προστασίας κάνουν διάκριση μεταξύ *πολιτικής* (τι πρέπει να προστατευτεί από ποιον) και *μηχανισμού* (πώς επιτυγχάνεται/υλοποιείται η πολιτική)
- η πολιτική μπορεί να διαφέρει από εγκατάσταση σε εγκατάσταση και από καιρού εις καιρόν στην ίδια εγκατάσταση
- πρέπει επίσης να γίνει διάκριση μεταξύ *προστασίας* (*protection*) και *ασφάλειας* (*security*) – η τελευταία αναφέρεται στο γενικότερο πρόβλημα της διατήρησης της ακεραιότητας του συστήματος και των δεδομένων του

Μηχανισμοί Προστασίας

Πεδία προστασίας (Protection Domains)

- ένα σύστημα υπολογιστή περιέχει πολλά **αντικείμενα (objects)** που πρέπει να προστατευτούν. Αυτά είναι:
 - αντικείμενα υλισμικού (ΚΜΕ, περιοχές της μνήμης, τερματικά, δίσκοι, εκτυπωτές κ.ά.)
 - αντικείμενα λογισμικού (σηματοφορείς, διεργασίες και τα (υπο)προγράμματά τους, αρχεία και βάσεις δεδομένων, κ.ά.)
- κάθε αντικείμενο έχει ένα μοναδικό όνομα (αναγνωριστικό ή ταυτότητα), μέσω του οποίου προσπελάζεται, κι ένα σύνολο πράξεων που μπορούν να εφαρμοστούν σε αυτό
- για παράδειγμα η ΚΜΕ επιτρέπει μόνο την εκτέλεση, οι περιοχές της μνήμης μπορούν να γραφούν και να διαβαστούν, οι εκτυπωτές μπορούν μόνο να τυπώσουν, οι σηματοφορείς μπορούν να τύχουν επεξεργασίας από τις πράξεις `wait`, `signal`, `sem_init` κ.ο.κ.
- ανάλογα με τα δικαιώματα που της έχουν δοθεί, κάθε διεργασία θα πρέπει να μπορεί να προσπελάσει μόνο ορισμένα αντικείμενα και να χρησιμοποιήσει μόνο ορισμένες πράξεις απ' αυτές που μπορούν να εφαρμοστούν σ' αυτά
- κάθε διεργασία πρέπει να εκτελείται μέσα σ' ένα ορισμένο *πεδίο προστασίας* που προσδιορίζει τους πόρους που μπορεί να προσπελάσει η διεργασία, καθώς επίσης και τους τρόπους με τους οποίους μπορεί αυτή να τους προσπελάσει

Μηχανισμοί Προστασίας

Πεδία προστασίας (Protection Domains)

- η ικανότητα εφαρμογής μιας πράξης ενός αντικειμένου ονομάζεται **δικαίωμα προσπέλασης ή πρόσβασης (access right)**
- ένα **πεδίο προστασίας (protection domain)** –γνωστό και ως *σφαίρα προστασίας (protection sphere)*, *δακτύλιος προστασίας (protection ring)*, *συμφραζόμενα (context)* και ως *καθεστώς προστασίας (protection regime)*– είναι ένα σύνολο διατεταγμένων **ικανοτήτων (capabilities)**, δηλαδή
 - ζευγών της μορφής (όνομα_αντικειμένου, σύνολο_δικαιωμάτων)
- το παρακάτω σχήμα δείχνει τρία πεδία προστασίας και τα ζεύγη που αυτά περιέχουν
- μία διεργασία εκτελούμενη στο πεδίο Π2 μπορεί να διαβάσει από το αρχείο X και να γράψει στο αρχείο Y
- μία διεργασία εκτελούμενη στο πεδίο Π1 μπορεί να εκτελέσει το αρχείο F1, να διαβάσει και ν' αλλάξει τα περιεχόμενα του αρχείου F2
- δύο πεδία προστασίας δεν είναι απαραίτητο να είναι ξένα μεταξύ τους π.χ. το ζεύγος (Y, {W}) ανήκει και στα δύο πεδία Π2 και Π3

$(F1, \{X\})$
 $(F2, \{RW\})$

Πεδίο Π1

$(X, \{R\})$
 $(Y, \{W\})$

Πεδίο Π2

$(F1, \{R\})$
 $(F3, \{RWX\})$

Πεδίο Π3

Μηχανισμοί Προστασίας

Πεδία προστασίας (Protection Domains)

- κατά τη διάρκεια της εκτέλεσής τους, οι διεργασίες μπορούν να αλλάζουν πεδία προστασίας
- οι κανόνες αλλαγής πεδίων διαφέρουν από σύστημα σε σύστημα
- στην απλούστερη μορφή το σύστημα μπορεί να διαθέτει δύο τρόπους λειτουργίας
- όταν μια διεργασία εκτελείται στην κατάσταση του επόπτη, το πεδίο προστασίας της είναι ολόκληρο το σύστημα (πεδίο επόπτη) και μπορεί να εκτελέσει όλες τις προνομιούχες εντολές
- όταν η διεργασία εκτελείται στην κατάσταση του χρήστη, το πεδίο προστασίας της είναι μόνο το πεδίο χρήστη και μπορεί να εκτελέσει μόνο τις μη προνομιούχες εντολές, κι έτσι μεταξύ άλλων μπορεί να προσπελάσει μόνο ένα μέρος της μνήμης του συστήματος
- αυτή η μορφή προστασίας, κοινή στους υπολογιστές των αρχών της δεκαετίας του '60, δεν είναι ικανοποιητική, διότι επιτρέπει είτε όλα τα προνόμια είτε κανένα από αυτά

Μηχανισμοί Προστασίας

Πεδία προστασίας (Protection Domains)

- μια πιο ευέλικτη μορφή προστασίας υλοποιήθηκε στους υπολογιστές που διέθεταν υλισμικό για την υποστήριξη τεμαχισμού
- παρόλο που ο τρόπος αυτός είναι καλύτερος από τον προηγούμενο, δεν επιτρέπει τη δυναμική αλλαγή των δικαιωμάτων που μια διεργασία έχει πάνω σε κάθε τμήμα της (εκτός αν η διεργασία μπορεί να εκτελεστεί στην κατάσταση του επόπτη)
- ο σωστός τρόπος είναι να δίνονται σε κάθε διεργασία *μόνο τα προνόμια που αυτή χρειάζεται να έχει για να εκτελέσει τις τρέχουσες απαιτήσεις της*
- αυτή η φιλοσοφία των *“ελάχιστων απαραίτητων προνομίων”* είναι η καλύτερη δυνατή, για ν’ αποτρέψει τις πράξεις λανθασμένων ή *“μοχθηρών”* (malicious) διεργασιών

Μηχανισμοί Προστασίας

Ειδικοί μηχανισμοί προστασίας στο UNIX

- κάθε διεργασία στο UNIX αποτελείται από δύο μέρη (δομές): το μέρος του χρήστη και το μέρος του πυρήνα ή συστήματος
- η αλλαγή του πεδίου προστασίας στο UNIX επιτυγχάνεται μέσω των κλήσεων του επόπτη
- όταν μια διεργασία εκτελέσει μία κλήση του επόπτη, τότε η διεργασία παύει να εκτελεί το τμήμα κειμένου της (και να προσπελάζει το τμήμα δεδομένων της) και εκτελεί το πρόγραμμα του πυρήνα (το οποίο προσπελάζει το τμήμα δεδομένων του συστήματος), που μπορεί να προσπελάσει όλη τη φυσική μνήμη, όλο το δίσκο κι όλους τους υπόλοιπους πόρους του συστήματος
- ο μηχανισμός προστασίας στο UNIX βασίζεται στα πεδία προστασίας κάθε αρχείου, που αντιστοιχούν στον ιδιοκτήτη του αρχείου, στην ομάδα (ή στις ομάδες) του ιδιοκτήτη και στους υπόλοιπους χρήστες
- κάθε πεδίο αποτελείται από τα τρία δυφία r , w και x , τα οποία ελέγχουν την ανάγνωση, εγγραφή και εκτέλεση των περιεχομένων του αρχείου αντίστοιχα
- το πεδίο προστασίας μιας διεργασίας ορίζεται από την ταυτότητα του χρήστη (uid) και την ταυτότητα της ομάδας (gid) του

Μηχανισμοί Προστασίας

Ειδικοί μηχανισμοί προστασίας στο UNIX

- δεδομένου οποιουδήποτε συνδυασμού (`uid`, `gid`) είναι δυνατό να απαριθμηθούν όλα τα αντικείμενα (αρχεία) –συμπεριλαμβανομένων και των περιφερειακών συσκευών, που θεωρούνται ειδικά αρχεία– που μπορούν να προσπελαστούν, καθώς και οι τρόποι (προνόμια) προσπέλασής τους
- δύο διεργασίες που έχουν το ίδιο ζεύγος (`uid`, `gid`) έχουν το ίδιο ακριβώς πεδίο προστασίας
- κάθε εκτελέσιμο αρχείο (πρόγραμμα) στο UNIX έχει δύο δυφία προστασίας, γνωστά ως `setuid` και `setgid` bits
- όταν μια διεργασία αποπειραθεί να εκτελέσει ένα πρόγραμμα (να εκτελέσει την κλήση `exec` σ' ένα αρχείο) του οποίου το δυφίο `setuid` ή το δυφίο `setgid` του είναι ίσο με 1, τότε, *αν της επιτρέπεται*, η διεργασία αυτή αποκτά ένα νέο `uid` και/ή ένα νέο `gid`
- αν το νέο ζεύγος (`uid`, `gid`) είναι αυτό του ιδιοκτήτη του αρχείου, τότε η διεργασία μπορεί να χρησιμοποιήσει το αρχείο με τον ίδιο τρόπο που αυτό μπορεί να χρησιμοποιηθεί από τον ίδιο τον ιδιοκτήτη του
- η δυνατότητα αυτή παρέχεται έτσι, ώστε οι διεργασίες των χρηστών να μπορούν να εκτελέσουν πράξεις που είναι κανονικά επιτρεπτές μόνο στον *υπερχρήστη* (*superuser*), όπως για παράδειγμα η δημιουργία ευρετηρίων

Μηχανισμοί Προστασίας

Ειδικοί μηχανισμοί προστασίας στο UNIX

- η εντολή `mkdir`, που δημιουργεί ένα ευρετήριο, χρησιμοποιεί την κλήση `mknod`, που μπορεί να χρησιμοποιηθεί μόνο από τον υπερχρήστη
- το αρχείο (η εντολή) `mkdir`, που χρησιμοποιεί την κλήση αυτή, ανήκει στον υπερχρήστη και έχει δυφία προστασίας 4755 (το πρώτο δυφίο παριστάνει το `setuid bit`) έτσι, ώστε οι διεργασίες των χρηστών να μπορούν να δημιουργούν ευρετήρια αλλά με πολύ περιορισμένο τρόπο
- με το νέο ζεύγος (`uid`, `gid`) η διεργασία έχει ένα νέο πεδίο προστασίας
- η εκτέλεση ενός προγράμματος (αρχείου) που έχει το `setuid` ή το `setgid bit` ίσο με ένα προκαλεί επίσης αλλαγή του πεδίου προστασίας της διεργασίας που εκτελεί το πρόγραμμα αυτό αν έχουν διαφορετικό ζεύγος, θα έχουν γενικά προσπέλαση σ' ένα διαφορετικό σύνολο αρχείων και προνομιών χρήσης τους
- είναι πολλές φορές χρήσιμο η διεργασία να μπορεί να γνωρίζει τις πραγματικές και τις νέες τιμές των `uid` και/ή `gid` της. Αυτό επιτυγχάνεται με τις κλήσεις `getuid` και `getgid`

Μηχανισμοί Προστασίας

Ειδικοί μηχανισμοί προστασίας στο UNIX

- οι κοινοί χρήστες δεν μπορούν να αλλάξουν τη `uid` τους, εκτός αν εκτελέσουν προγράμματα που έχουν το δυφίο `setuid` ίσο με 1
- ο υπερχρήστης μπορεί να εκτελέσει τις κλήσεις `setuid` και `setgid`, οι οποίες αλλάζουν τις τιμές των `uid` και `gid` ενός αρχείου
- ο υπερχρήστης μπορεί επίσης να αλλάξει τον ιδιοκτήτη ενός αρχείου με την κλήση `chown`
- η κλήση `umask` εφαρμόζει μια μάσκα δυφίων εσωτερικά στο σύστημα, η οποία κρύβει (αφήνει μη-δενικά) τα δυφία προστασίας ενός αρχείου, όταν αυτό δημιουργείται (μέσω των κλήσεων `creat` ή `mknod`)
- η μάσκα δυφίων κληρονομείται από τις διεργασίες παιδιά
- αν ένας φλοιός εκτελέσει την κλήση `umask` αμέσως μετά τη διεργασία `login`, καμία από τις διεργασίες του χρήστη δεν μπορεί να δημιουργήσει αρχεία που μπορούν να τροποποιηθούν από άλλους χρήστες
- είναι συχνά χρήσιμο να γνωρίζει ένα πρόγραμμα που να έχει το `setuid` bit ίσο με 1 αν ο χρήστης που το κάλεσε έχει πραγματικά την άδεια να χρησιμοποιήσει (προσπελάσει) ένα αρχείο, μ' άλλα λόγια αν η αρχική (πραγματική) `uid` του χρήστη έχει την άδεια αυτή
- αυτό επιτυγχάνεται με την κλήση `access`, η οποία –όπως και η κλήση `umask`– μπορεί να εκδοθεί απ' όλες τις διεργασίες

Μηχανισμοί Προστασίας

Γενικοί μηχανισμοί προστασίας

- ένας γενικός τρόπος μέσω του οποίου ένα λειτουργικό σύστημα μπορεί να προσδιορίσει ποια αντικείμενα ανήκουν σε ποια πεδία είναι η **μήτρα προσπέλασης (access matrix)**
- οι γραμμές της μήτρας αυτής παριστάνουν πεδία, ενώ οι στήλες της παριστάνουν αντικείμενα
- κάθε στοιχείο της μήτρας ορίζει το σύνολο των δικαιωμάτων προσπέλασης του αντίστοιχου αντικειμένου (από μια διεργασία, όταν η διεργασία αυτή εκτελείται) στο αντίστοιχο πεδίο
- μήτρα προσπέλασης, που αντιστοιχεί στο προηγούμενο σχήμα

Αντικείμενο Πεδίο	F1	F2	F3	X	Y
Π1	{X}	{RW}			
Π2				{R}	{W}
Π3	{R}		{RWX}		{W}

- η μήτρα αυτή είναι γενικά πολύ μεγάλη αλλά αραιή (sparse), διότι τα περισσότερα στοιχεία της είναι κενά
- ακόμη κι αν αποθηκευτούν μόνο τα μη κενά στοιχεία της, η μήτρα σπαταλάει πολύ χώρο, διότι δεν επιτρέπει εύκολα να ομαδοποιήσουμε τα αντικείμενα ή τα πεδία

- αν ένα αντικείμενο μπορεί να διαβαστεί απ' όλες τις διεργασίες, τότε θα πρέπει να υπάρχει ένα στοιχείο γι' αυτό σ' όλα τα πεδία

Γενικοί Μηχανισμοί Προστασίας

- Στην πράξη χρησιμοποιούνται δύο μέθοδοι αποθήκευσης της μήτρας: κατά στήλες ή κατά γραμμές

Λίστες Ελέγχου Προσπέλασης

- μέθοδος αποθήκευσης της μήτρας προσπέλασης κατά στήλες
- ισοδυναμεί με την αντιστοίχιση σε *κάθε αντικείμενο* μιας (ταξινομημένης) λίστας, που περιέχει όλα τα πεδία του και τους τρόπους με τους οποίους μπορούν αυτά να προσπελάσουν το αντικείμενο
- η λίστα αυτή είναι γνωστή ως **Λίστα Ελέγχου Προσπέλασης (Access Control List, ACL)**
- το σύστημα VAX/VMS χρησιμοποιούσε τη μέθοδο αυτή που παρέχεται επίσης κι από την Java
- κάθε ACL αποτελείται από μία ή περισσότερες *Καταχωρίσεις Ελέγχου Προσπέλασης (Access Control Entries, ACEs)* καμία από τις οποίες προσδιορίζει τα εξής:
 - i) το χρήστη ή την ομάδα των χρηστών που μπορεί να προσπελάσει το αντικείμενο
 - ii) τους τρόπους που επιτρέπεται η προσπέλαση του αντικειμένου (π.χ. RWED, όπου E για Execute και D για DELETE), και
 - iii) *εναλλακτικές επιλογές (options)*, για παράδειγμα στην περίπτωση που το στοιχείο (ACE) αναφέρεται σ' ένα ευρετήριο, αν μπορεί να ισχύει για όλα τα αρχεία του ευρετηρίου αυτού

Γενικοί Μηχανισμοί Προστασίας

Λίστες Ελέγχου Προσπέλασης

- μία ACL μπορεί να δοθεί σε κάθε αρχείο ή ευρετήριο
- η ταξινόμηση της λίστας ελέγχου προσπέλασης παίζει αρκετά σημαντικό ρόλο, διότι το ψάξιμό της αρχίζει από το πρώτο της στοιχείο
- ένας χρήστης (ή μια ομάδα χρηστών) μπορεί να εμφανίζεται σε περισσότερα από ένα στοιχεία της λίστας, και γενικά το πρώτο στοιχείο το αναφερόμενο στο χρήστη (ή στην ομάδα) προσδιορίζει τον τρόπο προσπέλασης που του (της) επιτρέπεται
- αν ένας χρήστης δεν εμφανίζεται καθόλου στη λίστα, τότε γι αυτόν ισχύει ένας προκαθορισμένος (default) τρόπος προσπέλασης
- αν ένας χρήστης επιχειρήσει προσπέλαση αντίθετη με αυτήν που προσδιορίζουν τα στοιχεία της λίστας, τότε αυτή του απαγορεύεται και προκαλείται μία συνθήκη εξαίρεσης (exception condition)
- για να ελαττωθεί ο χρόνος ψαξίματος της λίστας, είναι δυνατό να συγκρίνεται η προσπέλαση με το στοιχείο του προκαθορισμένου τρόπου πρώτα και με τα υπόλοιπα στοιχεία στη συνέχεια
- το κύριο πρόβλημα της μεθόδου που χρησιμοποιεί ACLs είναι ο χώρος που απαιτούν οι λίστες αυτές και η επιβράδυνση της προσπέλασης των αρχείων
- η μέθοδος αυτή είναι επιθυμητή μόνο στην περίπτωση που απαιτείται μεγάλη ασφάλεια

Γενικοί Μηχανισμοί Προστασίας

Λίστες Ικανοτήτων

- μέθοδος αποθήκευσης της μήτρας προσπέλασης κατά γραμμές ισοδυναμεί με την αντιστοίχιση σε *κάθε πεδίο* μιας λίστας, η οποία περιέχει τα αντικείμενα που μπορούν να προσπελαστούν, καθώς και τους τρόπους προσπέλασής τους
- η λίστα αυτή είναι γνωστή ως **Λίστα Ικανοτήτων (Capability List)** και κάθε στοιχείο της είναι γνωστό ως **ικανότητα (capability)**
- κάθε ικανότητα περιέχει τα εξής:
 - i) τον τύπο (του αντικειμένου) της
 - ii) τα δικαιώματα προσπέλασης του αντικειμένου αυτού, και
 - iii) το αναγνωριστικό, ταυτότητα/δείκτη του/προς το αντικείμενου/ο
- μία λίστα ικανοτήτων που αντιστοιχεί σ' ένα πεδίο μπορεί να προσπελαστεί *μόνο έμμεσα* από μια διεργασία που εκτελείται στο πεδίο αυτό
- για παράδειγμα, αν μία διεργασία θέλει να ζητήσει την εκτέλεση μιας πράξης πάνω σ' ένα ορισμένο αντικείμενο, τότε ζητάει από το σύστημα να της επιτρέψει να εκτελέσει την πράξη αυτή, δίνοντάς

του τη θέση (διεύθυνση) της ικανότητάς της μέσα στη λίστα ικανοτήτων του πεδίου και το δείκτη (ταυτότητα) του αντικειμένου ως παράμετρο

- η ζητούμενη προσπέλαση του αντικειμένου επιτρέπεται μόνο αν η διεργασία κατέχει (*possesses*) την απαιτούμενη ικανότητα

Γενικοί Μηχανισμοί Προστασίας

Λίστες Ικανοτήτων

- οι λίστες ικανοτήτων αποτελούν οι ίδιες αντικείμενα, που μπορούν να προσπελαστούν μέσω δεικτών από άλλες λίστες ικανοτήτων, διευκολύνοντας έτσι τον (κατα)μερισμό των (υπο)πεδίων
- οι λίστες ικανοτήτων (που είναι οι ίδιες αντικείμενα) πρέπει να προστατευτούν από τις διεργασίες των χρηστών. Για να επιτευχθεί αυτό, έχουν προταθεί δύο μέθοδοι
- η πρώτη μέθοδος είναι να αποθηκευτεί η λίστα ικανοτήτων της κάθε διεργασίας σ' ένα τμήμα που είναι προσπελάσιμο μόνο από το λειτουργικό σύστημα. Οι διεργασίες μπορούν ν' αναφερθούν στις ικανότητές τους μόνο έμμεσα όπως περιγράφηκε παραπάνω
- η δεύτερη μέθοδος είναι να έχει κάθε αντικείμενο ένα δυφίο-σημάδι (*tag*) που να διακρίνει αν αυτό είναι κοινό δεδομένο ή ικανότητα. Το δυφίο αυτό δεν είναι προσπελάσιμο από τις διεργασίες χρηστών, αλλά μόνο από το υλισμικό ή το υλισμικολογισμικό του λειτουργικού συστήματος. Παρόλο που ένα μόνο δυφίο αρκεί για τη διάκριση μεταξύ ικανοτήτων κι άλλων αντικειμένων, συχνά χρησιμοποιούνται κι άλλα δυφία, για να γίνει διάκριση μεταξύ των τελευταίων
- σε αντίθεση με τις λίστες ελέγχου προσπέλασης, οι λίστες ικανοτήτων συγκεντρώνουν όλες τις πληροφορίες που χρειάζονται για την εκτέλεση μιας διεργασίας

Γενικοί Μηχανισμοί Προστασίας

Λίστες Ικανοτήτων

- το λειτουργικό σύστημα αρκεί να ελέγξει την ικανότητα που του παρουσιάζει μία διεργασία και δε χρειάζεται να ψάχνει να βρει τη λίστα ελέγχου προσπέλασης του αντικειμένου, για να επιτρέψει ή όχι την προσπέλαση ενός αντικειμένου
- από την άλλη πλευρά οι λίστες ικανοτήτων δεν αντιστοιχούν άμεσα στις ανάγκες των χρηστών και παρουσιάζουν πρόβλημα στην περίπτωση *αφαίρεσης προνομίων*
- Στην περίπτωση *αφαίρεσης προνομίων (revocation)*, είναι δύσκολο για το σύστημα να βρει όλες τις ικανότητες τις αναφερόμενες σ' ένα αντικείμενο και να τις αφαιρέσει (ή μάλλον να τις “πάρει πίσω”), διότι αυτές βρίσκονται σε λίστες ικανοτήτων, που μπορεί να είναι αποθηκευμένες σ' οποιοδήποτε σημείο του δίσκου
- ένας τρόπος να λυθεί το πρόβλημα αυτό είναι να αποθηκεύονται μαζί με κάθε αντικείμενο δείκτες προς όλες τις ικανότητες που έχουν σχέση με αυτό. Ο τρόπος αυτός προξενεί μεγάλη επιβάρυνση χώρου
- ένας δεύτερος τρόπος να λυθεί το πρόβλημα είναι η κάθε ικανότητα να μην αναφέρεται άμεσα προς το ίδιο το αντικείμενο, αλλά ν' αναφέρεται σ' ένα έμμεσο αντικείμενο, που δείχνει προς το αντικείμενο. Στην περίπτωση αυτή το σύστημα αρκεί να κάνει το δείκτη του έμμεσου αντικειμένου ίσο με **nil** (null), για να αφαιρέσει το δικαίωμα χρήσης του αντικειμένου

Γενικοί Μηχανισμοί Προστασίας

Λίστες Ικανοτήτων

- ένας τρίτος τρόπος είναι να αντιστοιχίζεται σε κάθε ικανότητα μία ακολουθία δυφίων (ένα κλειδί). Το κλειδί αυτό ορίζεται όταν η ικανότητα αυτή δημιουργείται, και δεν μπορεί να τροποποιηθεί ούτε να εξεταστεί από τη διεργασία που κατέχει την ικανότητα
- ένα *κύριο κλειδί (master key)* αντιστοιχίζεται επίσης στο κάθε αντικείμενο
- όταν δημιουργείται η ικανότητα, τότε η τρέχουσα τιμή του κύριου κλειδιού αντιστοιχίζεται στην ικανότητα
- όταν χρησιμοποιείται το αντικείμενο (απαιτείται η ικανότητα), τότε το κύριο κλειδί του συγκρίνεται με αυτό της ικανότητας της διεργασίας. Αν τα δύο κλειδιά ταιριάζουν, τότε η προσπέλαση επιτρέπεται, ειδάλλως σημαίνεται μία συνθήκη εξαίρεσης
- η αφαίρεση των δικαιωμάτων στην περίπτωση αυτή γίνεται απλώς με αντικατάσταση της τιμής του κύριου κλειδιού του αντικειμένου από μία νέα τιμή
- η αντικατάσταση αυτή κάνει όλες τις προηγούμενες ικανότητες προσπέλασης του αντικειμένου να μην ισχύουν πια

Γενικοί Μηχανισμοί Προστασίας

Μηχανισμός Κλειδιού/Κλειδαριάς

- ο μηχανισμός αυτός είναι συνδυασμός των λιστών ελέγχου προσπέλασης και των ικανοτήτων. Κάθε αντικείμενο έχει μια λίστα διαφορετικών ακολουθιών δυφίων, που είναι γνωστές ως κλειδαριές (*locks*). Κάθε πεδίο έχει επίσης μία λίστα διαφορετικών ακολουθιών δυφίων, που είναι γνωστές ως κλειδιά. Ένα κλειδί ορίζεται όταν δημιουργείται μια ικανότητα
- μια διεργασία εκτελούμενη σ' ένα πεδίο μπορεί να προσπελάσει ένα αντικείμενο μόνο αν το πεδίο αυτό διαθέτει ένα κλειδί που ταιριάζει (είναι ίσο) με μια κλειδαριά του αντικειμένου
- η διαχείριση των κλειδιών ενός πεδίου γίνεται από το λειτουργικό σύστημα
- οι διεργασίες των χρηστών δεν πρέπει να μπορούν να εξετάζουν ή να τροποποιούν άμεσα τις λίστες των κλειδιών και (ιδιαίτερα των) κλειδαριών
- ο μηχανισμός κλειδιού/κλειδαριάς είναι ευέλικτος και οικονομικός, ιδιαίτερα αν το μήκος των κλειδιών/κλειδαριών επιλεγεί σχετικά μικρό
- τα κλειδιά μπορούν να μεταβιβάζονται ελεύθερα από πεδίο σε πεδίο
- επιπλέον, τα προνόμια προσπέλασης ενός αντικειμένου μπορούν εύκολα να αφαιρεθούν επιλεκτικά με την αλλαγή μερικών κλειδιών που αναφέρονται στο αντικείμενο

Γενικοί Μηχανισμοί Προστασίας

Μηχανισμός Κλειδιού/Κλειδαριάς

- ο μηχανισμός αυτός, τέλος, είναι ιδιαίτερα κατάλληλος για χρήση σε κατανεμημένα συστήματα

Γενικοί Μηχανισμοί Προστασίας

Δυναμικοί μηχανισμοί προστασίας

- στα προηγούμενα υποθέσαμε ότι η μήτρα προσπέλασης είναι στατική, δηλαδή ότι τα περιεχόμενά της δεν αλλάζουν με την πάροδο του χρόνου
- ακόμα και στην περίπτωση που το σύνολο των πόρων που μια διεργασία απαιτεί καθ' όλη τη διάρκεια της ζωής της παραμένει σταθερό, η στατική μήτρα δεν είναι ικανοποιητική, γιατί δίνει περισσότερα προνόμια απ' όσα χρειάζεται μια διεργασία σε ορισμένες φάσεις της εκτέλεσής της
- αν, για παράδειγμα, η διεργασία χρειάζεται να διαβάσει ένα αρχείο σε κάποια φάση της και αργότερα να γράψει στο αρχείο αυτό, τότε το πεδίο της στη στατική μήτρα θα πρέπει να περιέχει και τα δύο δικαιώματα
- έτσι πρέπει να επιτρέπουμε την αλλαγή των περιεχομένων της μήτρας προσπέλασης *δυναμικά*, όχι μόνο όταν νέα αντικείμενα δημιουργούνται, παλιά αντικείμενα διαγράφονται, οι ιδιοκτήτες αντικειμένων αποφασίζουν να δώσουν περισσότερα προνόμια στους άλλους χρήστες ή να αφαιρέσουν προνόμια από αυτούς, αλλά και κάθε φορά που μια διεργασία αλλάζει πεδίο
- η αλλαγή ενός πεδίου από μια διεργασία μπορεί να προστατευτεί, *αν τα πεδία θεωρηθούν αντικείμενα* και συμπεριληφθούν στα αντικείμενα (στήλες) της μήτρας προσπέλασης

Γενικοί Μηχανισμοί Προστασίας

Δυναμικοί μηχανισμοί προστασίας

- κατά τον ίδιο τρόπο οποιαδήποτε μεταβολή στα περιεχόμενα της μήτρας προσπέλασης μπορεί να γίνεται με ελεγχόμενο τρόπο, αν η ίδια η μήτρα προσπέλασης θεωρηθεί αντικείμενο – αν δηλαδή κάθε γραμμή της θεωρηθεί αντικείμενο
- το πρόβλημα της παροχής δυναμικής προστασίας απασχόλησε πολύ και εξακολουθεί να απασχολεί πολλούς ερευνητές της επιστήμης των υπολογιστών. Έχουν γίνει πολλές προτάσεις και μερικές από αυτές έχουν υλοποιηθεί σε μερικά (ως επί το πλείστον πειραματικά) λειτουργικά συστήματα
- οι προτάσεις αυτές διαφέρουν ως προς τον τύπο και το πλήθος των *πρωταρχικών* ή *πρωτογενών πράξεων* (*primitive operations*), που παρέχουν για την επεξεργασία ικανοτήτων, και ως προς το συνδυασμό των πράξεων αυτών σε *εντολές προστασίας*, που επιτρέπουν στους χρήστες να τις χρησιμοποιήσουν
- η θεωρητική έρευνα περιστρέφεται γύρω από το ερώτημα “αν είναι δυνατό να αποδειχτεί ότι ένα σύστημα που ξεκινάει από μια ασφαλής κατάσταση δεν μπορεί να καταλήξει σε μη ασφαλής κατάσταση με την εφαρμογή των πρωτογενών πράξεων”. Το πρόβλημα αυτό είναι παρόμοιο με το (γενικότερο) πρόβλημα της ασφαλούς υλοποίησης αντικειμένων σε μία γλώσσα προγραμματισμού

Γενικοί Μηχανισμοί Προστασίας

Δυναμικοί μηχανισμοί προστασίας

- και τα δύο προβλήματα μπορούν να απλοποιηθούν, αν εκφράζουμε τις ικανότητες όλων των αντικειμένων που πρέπει να τύχουν προστασίας, συναρτήσσει ικανοτήτων προστατευόμενων τμημάτων της μνήμης
- στην περίπτωση αυτή, οι ικανότητες όλων των αντικειμένων μπορούν να υλοποιηθούν μέσω του υλισμικού διαχείρισης της μνήμης
- το πρόβλημα της δυναμικής προστασίας μάς παρέχει ένα μηχανισμό για λύση του προβλήματος που προκύπτει όταν χρησιμοποιούνται παρακολουθητές που δεν επιτρέπουν στις διεργασίες χρηστών να δημιουργούν τους δικούς τους πόρους, να τους ελέγχουν ή να τους χρονοπρογραμματίζουν με το δικό τους τρόπο, ούτε επιτρέπουν την ταυτόχρονη προσπέλαση ενός πόρου από περισσότερες από μία διεργασίες
- για να λυθούν τα προβλήματα αυτά, πρέπει οι πόροι να βρίσκονται εκτός των παρακολουθητών και πρέπει να προστατευτούν με κάποιον άλλο τρόπο
- η προστασία κάθε πόρου μπορεί να επιτευχθεί μέσω μιας ειδικής διεργασίας, που ονομάζεται *διαχειριστής (manager)* ή *εξυπηρετητής (server)* του πόρου αυτού

▪ Γενικοί Μηχανισμοί Προστασίας

Δυναμικοί μηχανισμοί προστασίας

- για να λυθούν τα προβλήματα αυτά, πρέπει οι πόροι να βρίσκονται εκτός των παρακολουθητών και πρέπει να προστατευτούν με κάποιον άλλο τρόπο
- η προστασία κάθε πόρου μπορεί να επιτευχθεί μέσω μιας ειδικής διεργασίας, που ονομάζεται *διαχειριστής (manager)* ή *εξυπηρετητής (server)* του πόρου αυτού
- για να χρησιμοποιήσει μια διεργασία έναν πόρο, καλεί (στέλνει ένα μήνυμα) στο διαχειριστή (εξυπηρετητή), που απλώς της επιστρέφει μια *ικανότητα* χρήσης του πόρου. Όταν η διεργασία πάρει την ικανότητα, τότε πρέπει να την παρουσιάσει στο σύστημα για να προσπελάσει τον πόρο
- π.χ. για να χρησιμοποιήσει η διεργασία κάποιο αρχείο, ζητά την άδεια από ένα διαχειριστή αρχείων, για να χρησιμοποιήσει η διεργασία ένα γραμματοκιβώτιο, ζητά την άδεια από το διαχειριστή γραμματοκιβωτίων κ.ο.κ.
- το πρόβλημα της δυναμικής προστασίας μάς παρέχει ένα μηχανισμό για λύση του προβλήματος που προκύπτει όταν χρησιμοποιούνται παρακολουθητές που δεν επιτρέπουν στις διεργασίες χρηστών να δημιουργούν τους δικούς τους πόρους, να τους ελέγχουν ή να τους χρονοπρογραμματίζουν με το δικό τους τρόπο, ούτε επιτρέπουν την ταυτόχρονη προσπέλαση ενός πόρου από περισσότερες από μία διεργασίες

▪ Γενικοί Μηχανισμοί Προστασίας

Δυναμικοί μηχανισμοί προστασίας

- όταν η διεργασία τελειώσει τη χρήση του πόρου, τότε επιστρέφει την ικανότητα στο διαχειριστή που της την έδωσε
- ο διαχειριστής μπορεί τότε να δώσει την ικανότητα σε κάποια άλλη διεργασία σύμφωνα με το δικό του τρόπο χρονοπρογραμματισμού
- είναι φανερό ότι στην περίπτωση αυτή ο διαχειριστής πρέπει να έχει περισσότερα προνόμια από τις διεργασίες των χρηστών [ο όρος “ενίσχυση προνομίων” (*rights amplification*) χρησιμοποιείται για το σκοπό αυτό] και ότι θα πρέπει να υπάρχει η δυνατότητα δυναμικής (διακοπτόμενης, με εκτόπιση) αφαίρεσης των προνομίων από μια διεργασία που αρνείται να τα επιστρέψει

Διαχείριση Εργασιών (Job Management)

- μια εργασία αποτελείται από ένα σύνολο βημάτων και αντιμετωπίζεται ως ένα (ενιαίο) αντικείμενο για λογιστικούς σκοπούς
- όσον αφορά στο σύστημα, το κύριο χαρακτηριστικό μιας εργασίας είναι ότι είναι ανεξάρτητη απ' όλες τις υπόλοιπες εργασίες
- το σύστημα δημιουργεί μία ή περισσότερες διεργασίες για την εκτέλεση κάθε βήματος της εργασίας
- οι λόγοι για τους οποίους είναι επιθυμητό να αντιστοιχίσει το σύστημα περισσότερες από μία διεργασίες σε *μία εργασία* είναι:
 1. να επωφεληθούμε από τη δυνατότητα ύπαρξης περισσότερων από έναν φυσικών επεξεργαστών στο σύστημα
 2. να επωφεληθούμε από τη δυνατότητα παράλληλης επεξεργασίας μέσα στην εργασία
 3. να αποφύγουμε τους περιορισμούς του συστήματος ή να απλοποιήσουμε την οργάνωση των προγραμμάτων “σπάζοντας” την εργασία σε μικρότερες (δι)εργασίες, και
 4. να μπορούμε να χρησιμοποιήσουμε επανεισαγόμενα προγράμματα (αμιγείς διαδικασίες)
- η *ρίζα (root)* της ομάδας των διεργασιών (*process group*) που ανήκουν σε μία εργασία ή σ' ένα χρήστη ονομαζόταν στα μεν συστήματα δεσμίδων **οργανωτής της εργασίας (job organiser)** [ή ακόμη, σε μερικά συστήματα, *διευθυντής της εργασίας (job director)*], ενώ στα συστήματα πολλαπλής πρόσβασης ονομάζεται **διερμηνευτής των εντολών (command interpreter)** ή **φλοιός (shell)**

Διαχείριση Εργασιών

- η εικονική μηχανή που παρέχει ένα σύστημα στους χρήστες του εξαρτάται από τον οργανωτή της εργασίας που αυτό του διαθέτει
- πολλά συστήματα παρέχουν στους χρήστες τους περισσότερους από έναν οργανωτές εργασίας (φλοιούς), δηλαδή διαφορετικές εικονικές μηχανές υποστηριζόμενες από το ίδιο σύστημα
- οι οργανωτές αυτοί διερμηνεύουν *διαφορετικές γλώσσες ελέγχου εργασιών και/ή διαφορετικές γλώσσες εντολών* και παρέχουν διαφορετικές ευκολίες στους χρήστες
- το κοινό σημείο τους είναι ότι είναι υπεύθυνοι για τη δημιουργία και τη διαγραφή (είναι οι γονείς) των διεργασιών που εκτελούν τα βήματα της εργασίας και αντιμετωπίζουν τις συνθήκες –επιτυχίας ή αποτυχίας (εξαίρεσης)– της εκτέλεσης των διεργασιών αυτών (ή τις μεταβιβάζουν από τη μία διεργασία στην άλλη)
- υπάρχουν δύο τρόποι υλοποίησης των εντολών που υποβάλλουν ή πληκτρολογούν οι χρήστες:
- στον πρώτο τρόπο, ο διερμηνευτής των εντολών περιέχει αποσπάσματα που διερμηνεύουν με τον ίδιο τρόπο τις παραμέτρους όλων των εντολών
- ο τρόπος αυτός έχει το *μειονέκτημα* ότι το μέγεθος του οργανωτή είναι σχετικά μεγάλο και χρειάζεται να αλλαχτεί η δομή του, αν προστεθούν νέες εντολές ή αν αλλάξει ο τρόπος λειτουργίας των υπαρχόντων εντολών του λειτουργικού συστήματος

Διαχείριση Εργασιών

- στο δεύτερο τρόπο, οι παράμετροι διερμηνεύονται από τις διεργασίες που δημιουργούνται με την πληκτρολόγηση των εντολών αυτών στην περίπτωση αυτή, το πρόγραμμα του οργανωτή παραμένει μικρό και απλό, αλλά πρέπει κατά κάποιον τρόπο να μεταβιβάζει τις παραμέτρους των εντολών στις διεργασίες(π.χ. μέσω των παραμέτρων `argc` και `argv` στη C του UNIX)
- το *μειονέκτημα* του τρόπου αυτού είναι ότι η παροχή και διερμηνεία των παραμέτρων μπορεί να αντιμετωπίζεται με διαφορετικό τρόπο από τις διάφορες διεργασίες (π.χ. εργαλεία λογισμικού), που γράφηκαν από διαφορετικούς προγραμματιστές σε διάφορες φάσεις ανάπτυξης του συστήματος

Περιγραφητής εργασίας και ουρά εργασιών

- κάθε **εργασία** (κάθε χρήστης) που υπάρχει στο σύστημα έχει έναν **περιγραφητή (job descriptor)**, γνωστό και ως **Ομάδα Ελέγχου της Εργασίας (Job Control Block, JCB)**
- οι πληροφορίες που περιέχει ο περιγραφητής μιας εργασίας περιλαμβάνουν:
 - τον αριθμό (ή/και το όνομα) της εργασίας (του χρήστη)
 - πληροφορίες για τους πόρους που αυτή ξόδεψε
 - κ.ά.

Διαχείριση Εργασιών

Λογιστική εργασιών

- *λογιστική (accounting)* είναι η καταγραφή των ποσοτήτων που χρησιμοποιήθηκαν από κάθε πόρο του συστήματος
- η ιδανική πολιτική χρέωσης είναι να προσδιοριστούν οι ποσότητες των πόρων που θα χρησιμοποιούσε η εργασία αν βρισκόταν μόνη της στο σύστημα
- η εφαρμογή της πολιτικής αυτής εξασφαλίζει ότι η χρέωση της εργασίας δε θ' αλλάξει από την παρουσία άλλων εργασιών που μοιράζονται το σύστημα συγχρόνως με αυτήν
- μια άλλη άποψη είναι ότι το λειτουργικό σύστημα δουλεύει για λογαριασμό των χρηστών, και συνεπώς οι χρήστες θα πρέπει να χρεώνονται για την επιβάρυνση του λειτουργικού συστήματος
- η άποψη αυτή, αν εφαρμοστεί, έχει το πλεονέκτημα ότι ίσως να αποθαρρύνει τους χρήστες να γράφουν προγράμματα που αυξάνουν την επιβάρυνση του συστήματος, για παράδειγμα προγράμματα που εκτελούν άσκοπες προσπελάσεις αρχείων
- από την άλλη πλευρά, η επιβάρυνση του λειτουργικού συστήματος μεταβάλλεται ανάλογα με το φόρτο εργασίας που το σύστημα έχει σ' ένα ορισμένο χρονικό διάστημα

Διαχείριση Εργασιών

Λογιστική εργασιών

- έτσι είναι άδικο να χρεώνονται οι χρήστες περισσότερο, απλώς και μόνο διότι το σύστημα συμβαίνει να είναι (υπερ)φορτωμένο όταν διεκπεραιώνει την εργασία τους, ή διότι η εργασία τους έχει ανεπιθύμητες αλληλεπιδράσεις με ορισμένες άλλες εργασίες, που τυχαίνει να συνυπάρχουν στο σύστημα με αυτή
- επιπλέον, μπορεί να υποστηριχτεί ότι ο “κοινός χρήστης” δεν είναι υποχρεωμένος να αποκτήσει τη λεπτομερειακή (τεχνική) γνώση του λειτουργικού συστήματος που θα του χρειαστεί για την οργάνωση της εργασίας του με τέτοιο τρόπο, ώστε να ελαττώσει την επιβάρυνση του συστήματος
- η συμβιβαστική λύση είναι να χρεώνονται οι χρήστες για όλες τις δραστηριότητες του συστήματος που τους αφορούν αποκλειστικά (για παράδειγμα την είσοδο/έξοδο των δεδομένων που οι ίδιοι προκαλούν) και να μη χρεώνονται για τις δραστηριότητες που αυτοί δεν ελέγχουν (για παράδειγμα τη μεταγωγή των συμφραζομένων των διεργασιών τους, τη μεταφορά των τεμαχίων των διεργασιών τους από τη βοηθητική στην κύρια μνήμη και αντίθετα κ.ο.κ.)
- οι χρήστες δεν ενδιαφέρονται αν το σύστημα χρησιμοποιεί τεμαχισμό ή σελιδοποίηση για να τους προσφέρει την (εικονική) μνήμη για παράδειγμα
- το μόνο που βασικά τους ενδιαφέρει είναι να εκτελεστούν οι (δι)εργασίες τους

Διαχείριση Εργασιών

Λογιστική εργασιών

- ανεξάρτητα από την πολιτική χρέωσης των εργασιών, ο σχεδιαστής του συστήματος είναι υποχρεωμένος να παρέχει μηχανισμούς με τους οποίους:
 - (α) να μπορεί να μετρηθεί η χρήση των πόρων και να καταγραφεί σ' ένα λογιστικό αρχείο (*accounting file*) και/ή σ' ένα “ημερολόγιο του συστήματος” (*system's log*)
 - (β) να μπορούν να εξαχθούν από το αρχείο αυτό στατιστικά μεγέθη, που αφορούν στη χρήση των πόρων από κάθε χρήστη και από το ίδιο το σύστημα (παραγωγική εργασία + επιβάρυνση), και
 - (γ) να μπορούν να απορριφθούν οι χρήστες που έχουν ξεπεράσει τον προϋπολογισμό τους
- η μέτρηση της χρήσης των πόρων κατανέμεται σ' όλα τα επίπεδα χρονοπρογραμματισμού του συστήματος
- π.χ. στο επίπεδο του πυρήνα (βραχυχρόνιου χρονοπρογραμματιστή), στο επίπεδο του διαχειριστή των διεργασιών (μεσοχρόνιου χρονοπρογραμματιστή της ΚΜΕ) και στο επίπεδο του διαχειριστή της μνήμης (μεσοχρόνιου χρονοπρογραμματιστή της μνήμης)
- η παροχή επιπλέον καταχωρητών στο υλισμικό –με τη μορφή του χρονομέτρου διαστημάτων, των καταχωρητών ηλικίωσης και των μετρητών χρήσης (ή αναφοράς)– μπορεί να διευκολύνει τη συλλογή των μετρήσεων από το διαχειριστή της μνήμης

Διαχείριση Εργασιών

Λογιστική εργασιών

- κατά τον ίδιο τρόπο, οι ελεγκτές των δίσκων μπορούν να διαθέτουν ένα μετρητή-καταχωρητή, ο οποίος να μετράει το πλήθος των μεταφορών από ή προς αυτούς
- όλες αυτές οι μετρήσεις αποθηκεύονται στον περιγραφητή κάθε διεργασίας, συλλέγονται από το διαχειριστή της ΚΜΕ κάθε φορά που διαγράφεται μία διεργασία, και χρεώνονται στο γονέα της διεργασίας αυτής
- όταν ο οργανωτής της εργασίας (γονέας της ομάδας των διεργασιών) διαγραφεί, τότε οι μετρήσεις αποστέλλονται προς το διαχειριστή των εργασιών, για να ενημερώσει με τη σειρά του τα αντίστοιχα στοιχεία του περιγραφητή της εργασίας
- ο διαχειριστής των εργασιών χρειάζεται επίσης να καταγράψει τους χρόνους άφιξης και αναχώρησης των εργασιών (των χρηστών) και τις συνολικές μετρήσεις των πόρων που χρησιμοποίησαν στο λογιστικό αρχείο και στην οθόνη και/ή στις εκτυπώσεις των χρηστών

Διαχείριση Εργασιών

Λογιστική εργασιών

- η *τιμολόγηση (billing ή charging)*, δηλαδή το ποσό των χρημάτων που πρέπει να πληρώσουν οι χρήστες, δεν καθορίζεται από το σχεδιαστή του συστήματος, αλλά από τη διεύθυνση του γραφείου ή της εγκατάστασης που εξυπηρετεί (*service bureau ή installation*) τους χρήστες μέσω του ηλεκτρονικού υπολογιστή, σύμφωνα με κάποιο τύπο – που χρησιμοποιεί (συνήθως αυθαίρετα) *βάρη-τιμές* για να πολλαπλασιάζει τη *χρήση* κάθε πόρου και να τη μετατρέπει σε *χρήματα*
- πολλές εγκαταστάσεις υπολογιστών ανήκουν σε μία μόνο εταιρεία, ένα πανεπιστήμιο, έναν οργανισμό κ.ο.κ. και χρησιμοποιούνται αποκλειστικά από τα μέλη αυτών
- στην περίπτωση αυτή, η διεύθυνση δε χρεώνει τα μέλη της, αλλά έχει το πρόβλημα να ικανοποιήσει τις απαιτήσεις όλων των χρηστών, οι οποίες σχεδόν πάντα ξεπερνούν τις δυνατότητες της εγκατάστασης (ιδιαίτερα αν ο οργανισμός αυτός ανήκει στο ελληνικό δημόσιο και οι υπολογιστές λειτουργούν μόνο κατά τις “εργάσιμες ώρες”)
- στην περίπτωση αυτή, η διεύθυνση πρέπει να “καθορίζει τα μερίδια” (*rations*) των διαθέσιμων πόρων σύμφωνα με τις σχετικές ανάγκες ή τη “διάκριση” των χρηστών, δηλαδή χρειάζεται να εφαρμόσει κάποιον έλεγχο στη διανομή των πόρων

Διαχείριση Εργασιών

Έλεγχος Πόρων (Resource Control)

- ο έλεγχος των πόρων μπορεί να γίνει με το να περιοριστεί η προσπέλαση των χρηστών στους υπολογιστές
- επειδή αυτό μπορεί να προκαλέσει “παρεξηγήσεις”, είναι προτιμότερο να καθορίζονται προκαταβολικά τα μερίδια των πόρων που μπορεί να χρησιμοποιήσει κάθε χρήστης
- η πολιτική του καθορισμού των μεριδίων μπορεί να είναι βραχυχρόνια ή μακροχρόνια ή ένας συνδυασμός και των δύο
- η βραχυχρόνια πολιτική εφαρμόζεται στους πόρους του συστήματος που μπορεί να χρησιμοποιήσει μία εργασία κατά τη διάρκεια της εκτέλεσής της
- στη μακροχρόνια πολιτική δίνεται σε κάθε χρήστη κάποιος προϋπολογισμός που καθορίζει τους πόρους τους οποίους μπορεί να χρησιμοποιήσει ο χρήστης αυτός κατά τη διάρκεια κάποιας περιόδου, για παράδειγμα ενός μηνός ή ενός (ακαδημαϊκού) εξαμήνου
- αν ο χρήστης υπερβεί τον προϋπολογισμό του, τότε το σύστημα είτε αρνείται να τον δεχτεί είτε αποβάλλει (aborts) την εργασία του
- τα περισσότερα συστήματα πολλαπλής πρόσβασης παρέχουν μηχανισμούς μέσω των οποίων απαγορεύεται στους χρήστες να ξεπερνούν τα μερίδια του δίσκου που (ο διαχειριστής του συστήματος έχει αποφασίσει ότι) αναλογούν σ’ αυτούς (disk quota)

Διαχείριση Εργασιών

Έλεγχος Πόρων

- για παράδειγμα οι χρήστες του UNIX δεν μπορούν να υπερβούν το μέγιστο πλήθος των αρχείων και το μέγιστο (συνολικό) πλήθος των ομάδων που τους έχουν δοθεί από το διαχειριστή του συστήματος. Αυτό γίνεται ως εξής:
- κάθε στοιχείο του πίνακα των ανοικτών αρχείων μιας διεργασίας περιέχει μεταξύ άλλων την ταυτότητα του ιδιοκτήτη του αρχείου κι ένα δείκτη προς την εγγραφή ενός άλλου πίνακα, η οποία αντιστοιχεί στο χρήστη που άνοιξε το αρχείο αυτό
- ο πίνακας αυτός βρίσκεται επίσης στη μνήμη και περιέχει μια εγγραφή για κάθε χρήστη που έχει ανοίξει ένα ή περισσότερα αρχεία
- η εγγραφή αυτή προέρχεται από το *αρχείο των μεριδίων που αναλογούν (quota file)* σε κάθε χρήστη του συστήματος και επιστρέφει (γράφεται) στο δίσκο όταν ο χρήστης κλείσει όλα του τα αρχεία
- κάθε εγγραφή περιέχει οκτώ πεδία: “μαλακό όριο” (soft limit), “σκληρό όριο” (hard limit), τρέχον πλήθος και πλήθος των προειδοποιητικών μηνυμάτων που απομένουν (number of warnings left) – τέσσερα για τις ομάδες και τέσσερα για τα αρχεία του χρήστη
- κάθε φορά που προστίθεται μια ομάδα σ’ ένα αρχείο, το τρέχον πλήθος των ομάδων αυξάνεται κατά ένα και συγκρίνεται με το αντίστοιχο μαλακό και “σκληρό” όριο

Διαχείριση Εργασιών

Έλεγχος Πόρων

- κατά τη διάρκεια μιας *συνόδου (session)*, επιτρέπεται σ' ένα χρήστη να ξεπεράσει τα μαλακά όρια, αλλά απαγορεύεται να ξεπεράσει τα “σκληρά όρια”
- όταν ένας χρήστης επιχειρήσει να συνδεθεί με το σύστημα (login), ελέγχονται τα μαλακά όρια του πλήθους των ομάδων και των αρχείων που αντιστοιχούν σ' αυτόν
- αν οποιοδήποτε από αυτά έχει ξεπεραστεί, τότε το σύστημα τού δίνει ένα προειδοποιητικό μήνυμα και ελαττώνει το αντίστοιχο πλήθος των προειδοποιητικών μηνυμάτων που του απομένουν κατά 1
- αν τα δύο μαλακά όρια που απομένουν είναι ίσα με το μηδέν, τότε δεν επιτρέπεται στο χρήστη να συνδεθεί με το σύστημα
- με τον τρόπο αυτό, οι χρήστες μπορούν να ξεπεράσουν τα μαλακά τους όρια κατά τη διάρκεια μιας συνόδου τους, με την προϋπόθεση ότι θα διαγράψουν τις επιπλέον ομάδες ή αρχεία προτού αποσυνδεθούν από το σύστημα
- σε αντίθεση, τα σκληρά όρια δε γίνεται να ξεπεραστούν σε καμία περίπτωση
- ο καθορισμός των μεριδίων μπορεί να εφαρμοστεί ξεχωριστά για κάθε πόρο ή τα μερίδια μπορούν να πολλαπλασιαστούν με κατάλληλα βάρη και να προστεθούν, δίνοντας έτσι μία συνολική τιμή κατανάλωσης των πόρων

Διαχείριση Εργασιών

Έλεγχος Πόρων

- τα βάρη μπορούν να επιλεγούν με τέτοιο τρόπο, ώστε ν' αντικατοπτρίζουν τη “σπουδαιότητα” ή τη “σπανιότητα κάθε πόρου”
- ένας άλλος τρόπος είναι να δοθούν τέτοιες τιμές στα βάρη έτσι, ώστε να “ενθαρρύνουν τους χρήστες” να μη χρησιμοποιούν άσκοπα ορισμένους πόρους
- για παράδειγμα, αν δοθεί μεγάλη τιμή στο βάρος του χρόνου σύνδεσης των χρηστών με το σύστημα, οι “χρήστες θα ενθαρρυνθούν” να έχουν έτοιμα και καλογραμμένα τα προγράμματά τους, προτού συνδεθούν με τον υπολογιστή, και να αποφεύγουν να σπαταλούν άσκοπα τη διαθεσιμότητα του τερματικού “αυτοσχεδιάζοντας”
- είναι φανερό ότι ο καθορισμός των μεριδίων των πόρων μπορεί να είναι κατά τι μεγαλύτερος από τους πόρους τους οποίους μπορεί να διαθέσει το σύστημα, διότι οι περισσότεροι χρήστες συνήθως δε (θα πρέπει να) εξαντλούν όλον τον προϋπολογισμό τους

Διαχείριση Εργασιών

Έλεγχος Πόρων

- για να μπορεί να εφαρμόσει το σύστημα και ο διαχειριστής των εργασιών την πολιτική καθορισμού των μεριδίων των πόρων, το λογιστικό αρχείο πρέπει να περιέχει εγγραφές που να περιλαμβάνουν τα εξής:
 1. το όνομα ή τον αριθμό του (λογαριασμού) του χρήστη
 2. το (μυστικό) *συνθηματικό* (*password*) του, που χρησιμεύει για την εξακρίβωση της ταυτότητάς του (στα συστήματα αλληλεπίδρασης)
 3. ένα “*διάνυσμα άδειας*” (*permission vector*), κάθε δυφίο του οποίου παριστάνει την άδεια να χρησιμοποιήσει ο χρήστης ή η εργασία τον αντίστοιχο πόρο
 4. το μέγιστο ποσό κάθε πόρου που μπορεί να χρησιμοποιηθεί από την εργασία
 5. τον προϋπολογισμό των πόρων για την τρέχουσα περίοδο, και
 6. το υπόλοιπο που παραμένει σ’ αυτόν τον προϋπολογισμό για την τρέχουσα περίοδο

Διαχείριση Εργασιών

Δημιουργία διεργασιών

- αν ο μηχανισμός δημιουργίας των διεργασιών ανατεθεί εξ ολοκλήρου στο διαχειριστή των διεργασιών, τότε μπορεί να αυξηθεί ο “χρόνος απόκρισης” του διαχειριστή αυτού (που υποτίθεται ότι πρέπει να υπολογίζει τις “προτεραιότητες” των κρίσιμων διεργασιών ανά τακτά χρονικά (μεσο)διαστήματα)
- για το λόγο αυτό, είναι προτιμότερο να ανατεθεί ο μηχανισμός δημιουργίας των διεργασιών σε μία ξεχωριστή διεργασία ευρισκόμενη μεταξύ του διαχειριστή των διεργασιών και του διαχειριστή των εργασιών
- ανεξάρτητα από το επίπεδο που βρίσκεται η διεργασία αυτή (που παρέχεται ο μηχανισμός αυτός), η δημιουργία διεργασιών περιλαμβάνει τις παρακάτω ενέργειες:
- ο “δημιουργός διεργασιών” καλείται, όταν μία διεργασία-γονέας αποφασίζει να δημιουργήσει μία διεργασία-παιδί της. Όταν ο δημιουργός των διεργασιών κληθεί να δημιουργήσει μια διεργασία, πρέπει να κάνει τα εξής:

Διαχείριση Εργασιών

Δημιουργία διεργασιών

- (i) να ζητήσει από το διαχειριστή των διεργασιών να του καταχωρήσει έναν περιγραφητή για τη διεργασία αυτή και να κάνει την κατάσταση (του περιγραφητή) της διεργασίας αυτής *εμποδισμένη περιμένοντας για τη δημιουργία της*
Αν ο διαχειριστής των διεργασιών απαντήσει στο δημιουργό των διεργασιών ότι δεν μπορεί να του καταχωρήσει έναν περιγραφητή, τότε ο τελευταίος στέλνει ένα ανάλογο μήνυμα στο γονέα της διεργασίας
- (ii) να ζητήσει από το διαχειριστή των αρχείων να του ανοίξει το *αρχείο* (ή τα αρχεία) του *προγράμματος και των δεδομένων* της διεργασίας. Το αρχείο αυτό εκτός από τις (μεταθέσιμες) αντικειμενικές δομοενότητες –που αποτελούν το πρόγραμμα της διεργασίας– περιέχει και πληροφορίες για τα σημεία εισόδου τους, τη θέση και το μήκος κάθε δομοενότητας, τους αριθμούς των λογικών συσκευών (ρευμάτων) ή τα ονόματα των αρχείων που αυτή χρησιμοποιεί κ.ά.

Διαχείριση Εργασιών

Δημιουργία διεργασιών

(iii) να διαβάσει –μέσω του διαχειριστή των αρχείων ή του διαχειριστή της συσκευής που περιέχει τα αρχεία του προγράμματος και των δεδομένων της διεργασίας– τις πληροφορίες που χρειάζεται, και είτε να *συνδέσει στατικά* τις αντικειμενικές δομοενότητες σε μία φορτώσιμη δομοενότητα και να της καταχωρήσει χώρο στην περιοχή ανταλλαγής είτε να φορτώσει το *τμήμα σύνδεσης* της διεργασίας στην κύρια μνήμη, όταν η *σύνδεση* είναι *δυναμική*

Μερικές από τις πληροφορίες που χρειάζεται ο δημιουργός των διεργασιών μπορούν να έχουν πληκτρολογηθεί από το χρήστη ως παράμετροι των εντολών της γλώσσας (εντολών) που χρησιμοποιεί αυτός για την επικοινωνία του με το σύστημα

Αν η ανάγνωση των πληροφοριών αποτύχει, τότε η δημιουργία της διεργασίας δεν μπορεί να προχωρήσει

(iv) να ζητήσει από το διαχειριστή της μνήμης να κτίσει τον πίνακα των τεμαχίων της διεργασίας (ή έναν περιγραφητή για το τμήμα σύνδεσης της διεργασίας *πριν* το φορτώσει, όταν η σύνδεση γίνεται δυναμικά).

Αν ο διαχειριστής της μνήμης τον πληροφορήσει ότι αυτό δεν μπορεί να γίνει, τότε η δημιουργία της διεργασίας δεν μπορεί να προχωρήσει

Διαχείριση Εργασιών

Δημιουργία διεργασιών

- (v) να φορτώσει το κύριο τμήμα της διεργασίας και, αν το σύστημα χρησιμοποιεί πολιτική πρόβλεψης ή προσδοκίας, τότε να φορτώσει και τα τεμάχια του (προβλεπόμενου) συνόλου εργασίας της διεργασίας
- (vi) να ζητήσει από το εικονικό σύστημα εισόδου/εξόδου ή από το διαχειριστή των αρχείων να αντικαταστήσει τους αριθμούς των λογικών ρευμάτων ή των προκαθορισμένων αρχείων που χρησιμοποιεί η διεργασία, και να ορίσει κατάλληλα τα κανάλια της διεργασίας προς τους διαχειριστές των συσκευών αυτών και/ή προς το διαχειριστή των αρχείων, προς το γονέα της και προς το δημιουργό των διεργασιών (αν η διεργασία επιτρέπεται να δημιουργήσει διεργασίες-παιδιά)
- (vii) αν όλα πάνε κατ' ευχήν, ο δημιουργός των διεργασιών στέλνει ένα μήνυμα “δημιουργήθηκε” στο διαχειριστή των διεργασιών, για να αλλάξει την κατάσταση της διεργασίας σε έτοιμη και για να βάλει τον περιγραφητή της διεργασίας αυτής στην κατάλληλη θέση της ουράς του διανομέα. Ο δημιουργός των διεργασιών στέλνει επίσης ένα μήνυμα “να σου ζήσει” στο γονέα της διεργασίας

Διαχείριση Εργασιών

Δημιουργία διεργασιών

- αν όμως η δημιουργία της διεργασίας δεν μπορεί να προχωρήσει για οποιονδήποτε από τους παραπάνω λόγους, τότε ο δημιουργός των διεργασιών στέλνει ένα μήνυμα “απέτυχα” στο διαχειριστή των διεργασιών, για να αποδεσμεύσει τον περιγραφητή που αυτός έχει παραχωρήσει για τη διεργασία
- ο δημιουργός των διεργασιών στέλνει επίσης ένα μήνυμα “συλλυπητήρια” στο γονέα της διεργασίας, πληροφορώντας τον για το λόγο που απέτυχε η δημιουργία της διεργασίας-παιδιού του

Διαχείριση Εργασιών

Διαγραφή διεργασιών

- ο δημιουργός των διεργασιών δε χρειάζεται ν' αναμειχθεί στη διαγραφή τους
- όταν ένας γονέας αποφασίζει να διαγράψει μία διεργασία-παιδί, μπορεί να στείλει ένα μήνυμα “διάγραψε διεργασία” κατευθείαν στο διαχειριστή των διεργασιών
- στα περισσότερα συστήματα, ο διαχειριστής των διεργασιών αναλαμβάνει να διαγράψει όχι μόνο τη διεργασία αυτή αλλά κι όλα τα παιδιά της
- ο διαχειριστής των διεργασιών αλλάζει την κατάσταση κάθε διεργασίας που πρέπει να διαγράψει σε “ολοκληρωμένη ή εμποδισμένη περιμένοντας να θανατωθεί ή να διαγραφεί” και στέλνει ένα μήνυμα “διάγραψε τα τεμάχια της διεργασίας” στο διαχειριστή της μνήμης κι ένα μήνυμα στο διαχειριστή αρχείων για να ελευθερώσει την περιοχή ανταλλαγής που έχει καταχωρήσει στη διεργασία
- όταν ο διαχειριστής της μνήμης απαντήσει στο διαχειριστή των διεργασιών, τότε ο τελευταίος χρεώνει το γονέα της διεργασίας για όλους τους πόρους που έχει χρησιμοποιήσει η διεργασία αυτή, και αποδεσμεύει τον περιγραφητή της
- όταν διαγραφούν όλες οι διεργασίες που ανήκουν στο υποδέντρο της διεργασίας, τότε ο διαχειριστής της ΚΜΕ στέλνει ένα μήνυμα “εντάξει, τη διέγραψα” στο γονέα της διεργασίας και επιστρέφει στην κύρια είσοδό του.