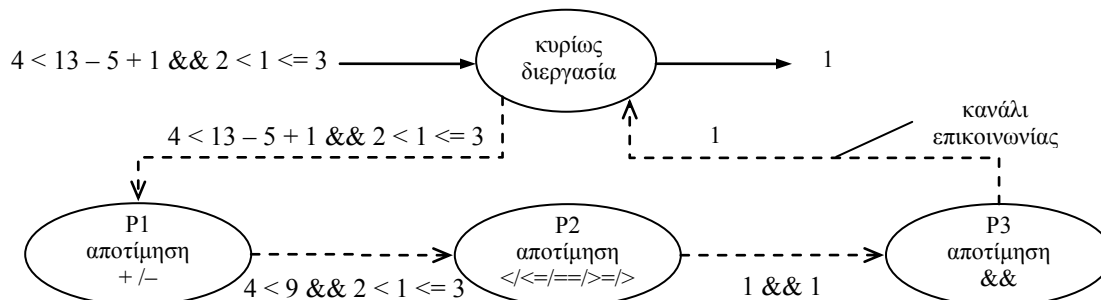


Εργασία 2 – Ροή επεξεργασίας με πολλές διεργασίες

Αναπτύξτε μια εφαρμογή που δέχεται λογικές εκφράσεις της μορφής “ $4 < 13 - 5 + 1 \ \&\& \ 2 < 1 \leq 3$ ” και εκτυπώνει το αποτέλεσμα (0 ή 1). Ο υπολογισμός πρέπει να υλοποιηθεί ως μια ροή επεξεργασίας με τρία στάδια καθένα από τα οποία εκτελείται μέσα από μια ξεχωριστή διεργασία και έτσι ώστε η προτεραιότητα των τελεστών να είναι πρώτα $+/-$, μετά $</<=//>=>$ και τέλος $\&\&$, στο πνεύμα του παρακάτω σχήματος:



Πιο συγκεκριμένα: (1) η κυρίως διεργασία διαβάζει την έκφραση από την είσοδο της, την στέλνει στην P1, λαμβάνει από την P3 το αποτέλεσμα, και το εκτυπώνει στην έξοδο της, (2) η P1 λαμβάνει την έκφραση που στέλνει η κυρίως διεργασία, εφαρμόζει τους τελεστές $+/-$, και στέλνει την μετασχηματισμένη έκφραση στην P2, (3) η P2 λαμβάνει την έκφραση που στέλνει η P1, εφαρμόζει τους τελεστές $</<=//>=>$, και στέλνει την μετασχηματισμένη έκφραση στην P3, και (4) η P3 εφαρμόζει τον τελεστή $\&\&$, και στέλνει το τελικό αποτέλεσμα στην κυρίως διεργασία. Η διαδικασία επαναλαμβάνεται για κάθε νέα έκφραση που δίνει ο χρήστης. Οι εκφράσεις «μεταδίδονται» ως ακολουθίες χαρακτήρων ASCII που τερματίζονται με $\backslash n$. Υποθέστε ότι οι εκφράσεις που δίνει ο χρήστης είναι συντακτικά σωστές, δηλαδή της μορφής $\langle \text{αριθμός} \rangle \{ \langle \text{τελεστής} \rangle \langle \text{αριθμός} \rangle \}$, μπορεί όμως να περιέχουν ενδιάμεσους λευκούς χαρακτήρες.

Κάθε στάδιο επεξεργασίας πρέπει να αναπτυχθεί ως ξεχωριστό πρόγραμμα που εκτελείται μέσα από μια διεργασία που δημιουργεί η κυρίως διεργασία κατά την εκκίνηση της εφαρμογής. Ο χρήστης ζητά τον τερματισμό της εφαρμογής μέσω της ειδικής έκφρασης 'q' , που στέλνεται κατά μήκος της γραμμής επεξεργασίας, με κάθε διεργασία να τερματίζει αφού πρώτα προωθήσει το 'q' στην επόμενη. Τελευταία τερματίζει η κυρίως διεργασία, αφού λάβει το 'q' και βεβαιωθεί ότι τερματίστηκαν όλες οι υπόλοιπες διεργασίες P1, P2 και P3.

Επικοινωνία με αρχεία: Υλοποιήστε κάθε κανάλι επικοινωνίας μέσω ενός αρχείου. Το μέγεθος του αρχείου υποδηλώνει κατά πόσο το κανάλι είναι έτοιμο για αποστολή ή παραλαβή δεδομένων. Συγκεκριμένα, για να στείλει μια διεργασία δεδομένα, περιμένει μέχρι το μέγεθος του αρχείου να γίνει 0 και στη συνέχεια γράφει τα δεδομένα στο αρχείο. Για να παραλάβει μια διεργασία δεδομένα, περιμένει μέχρι το μέγεθος του αρχείου να γίνει > 0 , διαβάζει τα περιεχόμενα του αρχείου, και μηδενίζει το μέγεθος του αρχείου. Το κυρίως πρόγραμμα πρέπει να δημιουργεί τα αρχεία κατά την εκκίνηση, και να τα σβήνει πριν τον τερματισμό. Τα υπόλοιπα προγράμματα πρέπει να δέχονται ως ορίσματα τα ονόματα των αρχείων που θα χρησιμοποιήσουν ως κανάλια επικοινωνίας.

Επικοινωνία με αγωγούς: Αναπτύξτε μια διαφορετική έκδοση της εφαρμογής (όλων των προγραμμάτων) όπου τα κανάλια επικοινωνίας ανάμεσα στις διεργασίες υλοποιούνται με ανώνυμους αγωγούς, και έτσι ώστε οι P1, P2 και P3, να λαμβάνουν και να στέλνουν δεδομένα διαβάζοντας από την συμβατική είσοδο και γράφοντας στην συμβατική έξοδο τους, αντίστοιχα. Η κυρίως διεργασία πρέπει να δημιουργεί τους αγωγούς και να ανακατευθύνει κατάλληλα την είσοδο και έξοδο των διεργασιών P1, P2 και P3.

Και τα δύο μέρη της εργασίας είναι υποχρεωτικά. Για την πρόσβαση σε αρχεία και την είσοδο/έξοδο των προγραμμάτων σας μπορείτε να χρησιμοποιήσετε (και) τις λειτουργίες της βιβλιοθήκης `stdio` (όμως προσέξτε την μικτή χρήση συναρτήσεων της `stdio` με τις κλήσεις συστήματος `read/write/ftruncate/truncate`). Απαγορεύονται οι καθολικές (global) μεταβλητές και η χρήση `goto`. Τεκμηριώστε κάθε μεταβλητή και συνάρτηση του προγράμματος. Ο κώδικας σας πρέπει να είναι ευανάγνωστος. Κώδικας που «δεν διαβάζεται» θα απορριφθεί, χωρίς εξέταση.

Παράδοση: Τρίτη 1 Απριλίου 2014, 22:00

Οδηγίες Παράδοσης: Στην ιστοσελίδα του μαθήματος