

## Προγραμματιστική εργασία C – Processes, Pipes, Signals

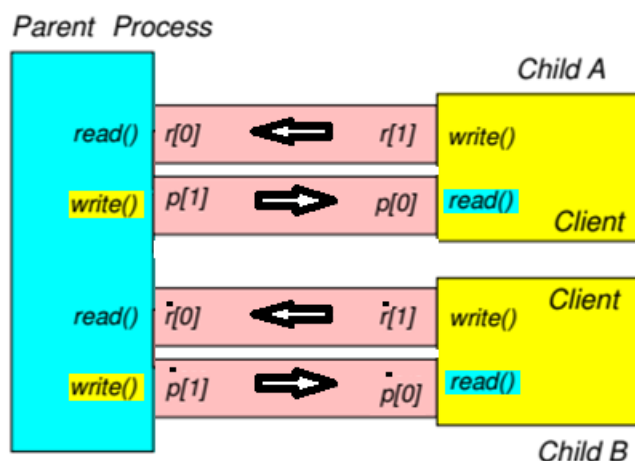
### Έκδοση 1.0

#### Αντικείμενο εργασίας

Σκοπός της εργασίας είναι να δημιουργήσουμε ένα πρόγραμμα σε γλώσσα προγραμματισμού C όπου θα υλοποιούνται 3 διεργασίες όπου η μία θα έχει το ρόλο του δρομολογητή (scheduler) υπολογιστικών εργασιών, ενώ οι άλλες δύο θα έχουν τον ρόλο να πραγματοποιήσουν τις υπολογιστικές εργασίες (workers) που θα τους αναθέτει η διεργασία scheduler.

Καλείστε να δημιουργήσετε ένα πρόγραμμα που θα γεννάει δύο διεργασίες παιδιά (child processes) που θα παίξουν το ρόλο των workers ενώ η κύρια διεργασία που τα γέννησε (parent process) θα παίξει το ρόλο του scheduler εργασιών.

Η parent process επικοινωνεί με την κάθε μια από τις child process με δύο μονόδρομα κανάλια επικοινωνίας (unidirectional channels) που το ένα χρησιμοποιείται μόνο για εγγραφή (write) δεδομένων από την parent process και διάβασμα (read) από την child process, ενώ το άλλο για το αντίθετο - εγγραφή (write) δεδομένων από την child process και διάβασμα (read) από την parent process. Τα μονόδρομα κανάλια επικοινωνία υλοποιούνται με την χρήση σωληνώσεων (pipes). Για κάθε child process θα χρειαστεί να έχετε δύο μονόδρομες σωληνώσεις για να υλοποιήσετε την επικοινωνία με την parent process.



Τα κανάλια επικοινωνίας που θα δημιουργήσετε ανάμεσα στην parent process και τις child process θα χρησιμοποιηθούν για να υποβάλει υπολογιστικές εργασίες η parent process προς τις child process ενώ οι τελευταίες για να επιστρέφουν το αποτέλεσμα της υπολογιστικής εργασίας στην parent process.

Η parent process έχει μόνο τον ρόλο του scheduler εργασιών και η ίδια δεν θα εκτελεί υπολογιστικές εργασίες.

Όταν η parent process έχει εγγράψει στη κατάλληλη σωλήνωση (pipe) την εργασία που θέλει να υποβάλει στο πρώτο child process του στέλνει ένα σήμα (signal) SIGUSR1, ενώ όταν έχει εγγράψει στη κατάλληλη σωλήνωση (pipe) την εργασία που θέλει να υποβάλει στο δεύτερο child process του στέλνει ένα σήμα SIGUSR2. Όταν κάποιο από τα child process λάβει το σήμα που προαναφέραμε τότε διαβάζει την εργασία από τη σωλήνωση και πραγματοποιεί τους υπολογισμούς.

Από τη μεριά τους οι child process όταν ολοκληρώσουν την υπολογιστική εργασία που τους έθεσε η parent process στέλνουν σε αυτή ένα σήμα (signal) – SIGUSR1 αν πρόκειται για το πρώτο child process και SIGUSR2 αν πρόκειται για το δεύτερο child process. Λαμβάνοντας το σήμα η parent process διαβάζει την κατάλληλη σωλήνωση και παίρνει το αποτέλεσμα του υπολογισμού που υπέβαλε.

Θα χρησιμοποιήσουμε αυτή τη δομή για να εκτελούμε αριθμητικές πράξεις στα child process. Η parent process προκειμένου να υποβάλει μια εργασία αριθμητικού υπολογισμού εγγράφει στην κατάλληλη σωλήνωση πρώτα τον 1<sup>ο</sup> αριθμό μετά το τελεστή (+, -, \*, /, ^) και μετά τον 2<sup>ο</sup> αριθμό – κάνει τρεις εγγραφές. Στέλνει έπειτα το κατάλληλο σήμα (signal) για να ειδοποιήσει το child process ότι του ανέθεσε μια εργασία. Το child process διαβάζει την κατάλληλη σωλήνωση και εκτελεί την πράξη. Εγγράφει το αποτέλεσμα στη κατάλληλη σωλήνωση και στέλνει σήμα (signal) στη parent process για να διαβάσει το αποτέλεσμα της υπολογιστικής πράξης που του ανέθεσε από την κατάλληλη σωλήνωση.

#### A. 80% της βαθμολογίας

Έχουμε να υπολογίσουμε το ακόλουθο:  $X = (2*7 + 3*6 - 2^2 + 3*3 - 2)/2$

Χρησιμοποιείτε τη δομή που δημιουργήσατε προκειμένου να κάνετε τον υπολογισμό αφού σπάσετε κατάλληλα το παραπάνω σε μια σειρά πράξεων που θα υποβάλετε προς τις child process να εκτελέσουν. Η parent process έχει τον ρόλο να δρομολογεί τους υπολογισμούς και να λαμβάνει τα αποτελέσματα μέχρι να έχει το τελικό αποτέλεσμα X το οποίο και θα παρουσιάσει στην οθόνη. Η parent process δεν θα εκτελέσει κανένα υπολογισμό. Τους υπολογισμούς προσπαθήστε να τους υποβάλετε όσο το δυνατό πιο ομοιόμορφα και στα δύο child process ώστε και τα δύο child process – workers να εκτελούν εργασίες ταυτόχρονα.

Παράγετε ένα πρόγραμμα με όνομα `shed_two_workers.c`

## A. 20% της βαθμολογίας

Τροποποιήστε τον τρόπο που μαθαίνει τόσο το parent process όσο και τα child process αν έχει εγγραφεί κάτι στην κατάλληλη σωλήνωση (ώστε να κάνει τις κατάλληλες ενέργειες), χρησιμοποιώντας την κλήση συστήματος select και όχι τα σήματα (signals SIGUSR1 και SIGUSR2) που χρησιμοποιήσαμε προηγουμένως.

Παράγετε ένα νέο πρόγραμμα με όνομα shed\_two\_workers\_ns.c

*Σε όλα τα προγράμματα που θα παράγετε ακολουθήστε όλες τις ενδεδειγμένες τακτικές για να παράγετε κώδικα που θα είναι κατάλληλα δομημένος και **με σχόλια** που θα τον συνοδεύουν, ενώ το παραγόμενο πρέπει να είναι φιλικό στον χρήστη παρέχοντας τις κατάλληλες οδηγίες στο command line.*

### **Παραδοτέα**

- Μια σύντομη και περιεκτική εξήγηση για τις επιλογές που έχετε κάνει στο σχεδιασμό του προγράμματός σας καθώς και αναφορά στις συγκρίσεις που μπορεί να σας ζητούνται.
- Ο κώδικας που θα υποβάλετε θα πρέπει να είναι δικός σας. Δεν επιτρέπεται η χρήση κώδικα που δεν έχει γραφεί από εσάς (αυτό συμπεριλαμβάνει και κώδικα από το Διαδίκτυο).
- Όλη η δουλειά σας (πηγαίος κώδικας και README) σε ένα tar.gz file με ονομασία OnomaEponymoProject2.tar.gz.
- Κρατήστε ένα backup .tar της άσκησής σας όπως ακριβώς αυτή υποβλήθηκε.
- Η σωστή υποβολή ενός σωστού tar.gz που περιέχει τον κώδικα της άσκησής σας και ότι αρχεία χρειάζονται είναι αποκλειστικά ευθύνη σας.

### **Διαδικαστικά**

- Βεβαιωθείτε πως ακολουθείτε καλές πρακτικές software engineering κατά την υλοποίηση της άσκησης. Η οργάνωση, η αναγνωσιμότητα και η ύπαρξη σχολίων στον κώδικα αποτελούν κομμάτι της βαθμολογίας σας.
- Η υποβολή θα γίνει μέσω του eclass. Θα ειδοποιηθείτε σχετικά όταν ξεκινήσει η υποβολή των εργασιών. Η υποβολή θα είναι δυνατή μέχρι την ημέρα και ώρα που αναφέρεται στη προθεσμία υποβολής.

### **Άλλες σημαντικές παρατηρήσεις**

- Οι εργασίες είναι ατομικές.
- Αν και αναμένεται να συζητήσετε με φίλους και συνεργάτες το πώς θα επιχειρήσετε να δώσετε λύση στο πρόβλημα, αντιγραφή κώδικα (οποιασδήποτε μορφής) είναι κάτι που δεν επιτρέπεται και ελέγχεται με αυτοματοποιημένο και μη τρόπο.
- Θα πρέπει να λάβετε τα κατάλληλα μέτρα ώστε να είναι προστατευμένος ο κώδικάς σας και να μην αποθηκεύεται κάπου που να έχει πρόσβαση άλλος χρήστης.
- Ενδέχεται να κληθείτε να παρουσιάσετε και να εξεταστείτε στις εργασίες σας αναλύοντας τη προσέγγισή σας και τις προγραμματιστικές σας επιλογές.
- Οι ασκήσεις προγραμματισμού πρέπει να δοθούν μέχρι την καταληκτική ημερομηνία.