

Προγραμματισμός Συστηματος

Εργασία 2^η

Όνομα:Μελίνα Μαρία Βλαχου

AM:1115201500017

Η εργασία είναι υλοποιημένη σε c++.

Compile: make

Run: ./mirror_client -n id -c common_dir -i input_dir -m mirror_dir -b buffer_size -l log_file

Περιγραφή της εφαρμογής:

Ξεκινώντας η main ορίζει και προσαρμόζει όλους τους handlers (SIGALRM, SIGINT, SIGQUIT, SIGCHLD, SIGUSR1, SIGUSR2) και στη συνέχεια ελέγχει την ύπαρξη ή μη των φακέλων που δίνονται. Γράφει το id στο logfile για τα στατιστικά αργότερα και στέλνει στον εαυτό της ένα SIGALRM το οποίο από δω και πέρα θα στέλνεται επαναληπτικά ανά DURATION χρόνο, ο οποίος καθορίζεται με define στην αρχή.

Μεσα στον handler SIGALRM αρχικά καλείται η συνάρτηση **scanChanges** η οποία ελέγχει το common directory για αλλαγές: την εμφάνιση ενός καινούριου client ή την ξαφνική απουσία ενός υπάρχοντος. Ενημερώνει με τις αλλαγές αυτές τις λίστες inserted και deleted. Στη συνέχεια, για κάθε νέο client (κομβό της λίστας inserted) καλείται η **createChilds** η οποία γεννά 2 παιδιά με fork έτσι ώστε να ξεκινήσει η επικοινωνία με τους άλλους clients.

Το 1^ο παιδί (ο αποστολέας) δημιουργεί ένα named pipe(fifo) και καλεί την **sendAllFiles**. Η συνάρτηση αυτή ανοίγει το input που έχει δοθεί καλείται αναδρομικά για ακαθε άλλο φάκελο που βρίσκεται μέσα στον αρχικό ενώ για κάθε αρχείο καλεί την **sendFile** η οποία στέλνει μέσω του pipe το μέγεθος του ονόματος του αρχείου, μετά το όνομα του αρχείου, το μέγεθος του αρχείου και τέλος ολόκληρο το αρχείο με τον εξής τρόπο: ανοίγει το αρχείο και διαβάζει κάθε φορά buffersize bytes ή αν απομένουν λιγότερα, διαβάζει όσα απομένουν. Στη συνέχεια καλεί την **writeAll** η οποία γράφει στο pipe χρησιμοποιώντας την write. Αν η write επιστρέψει κάτι αρνητικό ανιχνεύεται το λάθος. Μετά από αυτό η sendFile αν δεν έχει υπάρξει λάθος γράφει στο logfile την μεταφορά και επιστρέφει το ίδιο και η sendAllFiles και είναι γνωστό αν υπήρξε λάθος. Πίσω στην createChilds τώρα εάν δεν υπήρξε λάθος στέλνεται το 00 μέσω του pipe το οποίο κλείνει και τερματίζει με exit status 0 (στέλνεται SIGCHLD στον πατέρα) ενώ αν υπήρξε λάθος το παιδί τερματίζει με exit status το id του client με τον οποίο απέτυχε η αποστολή αρχείων.

Το 2^ο παιδι (ο παραληπτής) ακολουθεί αντιστοιχη διαδικασία για την αναγνώση αρχειων από το pipe όμως πριν ξεκινήσει να διαβάζει (**εφόσον χρησιμοποιούνται blocking pipes**) ξεκινάει η διαδικασία του rolling με την οποία περιμένει μέχρι και 30 δευτερόλεπτα να εμφανιστεί απόστολεας. Αν περάσουν και δεν εμφανιστεί τερματίζει και στέλνει SIGUSR1 στον γονέα. Αν εντοπίσει λάθος κατά την αναγνώση στέλνει πρώτα SIGUSR2 στον γονέα και μετά επίσης τερματίζει με exit status το id του client με τον οποίο απέτυχε η αποστολή αρχειων. Αν δεν εντοπίσει τερματίζει με exit status 0 (στέλνεται SIGCHLD στον πατέρα)

Στη συνέχεια ο SIGALRM handler για κάθε διεγραμενη εγγραφή γεννά ένα παιδι το οποίο διαφραφεί από το mirror του το φακέλο του client που αποχώρησε μέσω της αναδρομικής **myrmDir** και τερματίζει(εδώ στέλνεται επίσης SIGCHLD στον γονέα)

Όταν λαμβάνεται SIGINT ή SIGQUIT διαγραφεται το mirror directory, διαφραφεται το αρχείο του client από το common directory γραφεται στο logfile ένα συμβολο L(left) και τερματίζει.

Σε περίπτωση που ο γονέας λάβει το σήμα SIGUSR2, τότε αναζητάει στη λίστα known το pid του παιδιού που έστειλε το signal (μέσω της waitpid) και αν ο μετρητής αποτυχιών του συγκεκριμένου παιδιού δεν ξεπερνάει την τιμή 3, το ξεκινάει πάλι από την αρχή.

Τα get_stats.sh και create_infiles.sh έχουν υλοποιηθεί όπως έχει ζητηθεί απ την εκφώνηση.

```
Run:  chmod 755 create_infiles.sh
      ./create_infiles.sh dirname 15 5 3

      chmod 755 get_stats.sh
      cat log1 log2 | ./get_stats.sh
```