

Δοκιμάστε να αλλάξετε τα SIGUSR1 και SIGUSR2 σήματα με τα SIGRTMIN+1 και SIGRTMIN+2 σήματα αντίστοιχα. Παρατηρείτε κάποια διαφορά? Αν ναι που πιστεύετε ότι οφείλετε? Γιατί sigaction δεν αρκεί σε κάποιες περιπτώσεις?

Η διαφορά που παρατηρώ είναι ότι όταν χρησιμοποιώ τα SIGRTMIN+1 SIGRTMIN+2 χάνονται λιγότερα σήματα και αυτό συμβαίνει γιατί αυτά τα σήματα ανήκουν στην κατηγορία real time signals . Αυτά τα σήματα όταν φθάνουν ταυτόχρονα σε μια διεργασία μπαίνουν σε μια ουρά με αποτέλεσμα να μην χάνονται. Ενώ τα πρώτα σήματα όταν έρχονται ταυτόχρονα η διεργασία λαμβάνει υπόψη μόνο ένα και αγνοεί τα υπόλοιπα.

Παρολα αυτά πάλι μερικές φορές τα σήματα αγνοούνται και αυτό συμβαίνει διότι η sigaction την στιγμή που διαχειρίζεται ένα σήμα θα αγνοήσει τα υπόλοιπα που θα έρθουν.

Στην εκτέλεση του προγράμματος φαίνεται ο αριθμός σημάτων που δέχομαι.

Η hash function που χρησιμοποιώ είναι απο εδώ

<http://algoviz.org/OpenDSA/Books/OpenDSA/html/HashFuncExamp.html#StringSfold>

Μετέτρεψα τον κώδικα ώστε να ταιριάζει στην άσκηση και από κάτω αναφέρεται γιατί είναι καλή hash.

This function takes a string as input. It processes the string four bytes at a time, and interprets each of the four-byte chunks as a single long integer value. The integer values for the four-byte chunks are added together. In the end, the resulting sum is converted to the range 0 to M-1 using the modulus operator.

For example, if the string "aaaabbbb" is passed to sfold, then the first four bytes ("aaaa") will be interpreted as the integer value 1,633,771,873, and the next four bytes ("bbbb") will be interpreted as the integer value 1,650,614,882. Their sum is 3,284,386,755 (when treated as an unsigned integer). If the table size is 101 then the modulus function will cause this key to hash to slot 75 in the table.

Επίσης πειραματικά όταν το έτρεξα είδα ότι κάνει καλό διαμερισμό και αυτό φαίνεται και στο terminal .

Περιγραφή προγράμματος.

Δημιουργώ τρία κύρια προγράμματα (δηλαδή με main) splitter root builder

Αρχικά βασικά εκτελείτε ο root που με fork και execvp καλεί όσες φορές ζητείτε από τα ορίσματα τους splitter και τους builder.

Ο root παίρνει από το αρχείο που διαβάζει πόσες γραμμές θα είναι το κείμενο το διαιρεί με τους splitters και το περνάει σαν όρισμα . Σε περίπτωση που δεν είναι ακριβές πολλαπλασίο τότε ολοι οι

σπλιτερ παίρνουν το αποτέλεσμα της διαρεσης και ο τελευταίος splitter παίρνει τα υπόλοιπα για παραδειγμα αν έχω 10 γραμμες και 3 splitter οι δύο θα πάρουν από τρεις γραμμες και ο τελυταιος τις υπόλοιπες δηλαδή 4.

Οι λέξεις από το exclusive αρχείο αποθηκεύονται και αυτές σε μια λίστα και οταν εισάγω μια λέξη απο το textfile θα μπει σε μια άλλη λίστα αφού δεν υπάρχει στη exclusive.

Σαν χρονισμό στον root χρησιμοποιώ το παράδειγμα του κύριου Δελλη από την εκφώνηση.

Για το gnuplot πάλι χρησιμοποιώ ένα scriptaki και με exec το καλώ.