ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΘΝΙΚΌ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΌ ΠΑΝΕΠΙΣΤΉΜΙΟ ΑΘΗΝΏΝ
ΤΜΉΜΑ ΠΛΗΡΟΦΟΡΙΚΉΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΏΝ

Τίτλος Εργασίας

1η Προγραμματιστική Εργασία

Μάθημα

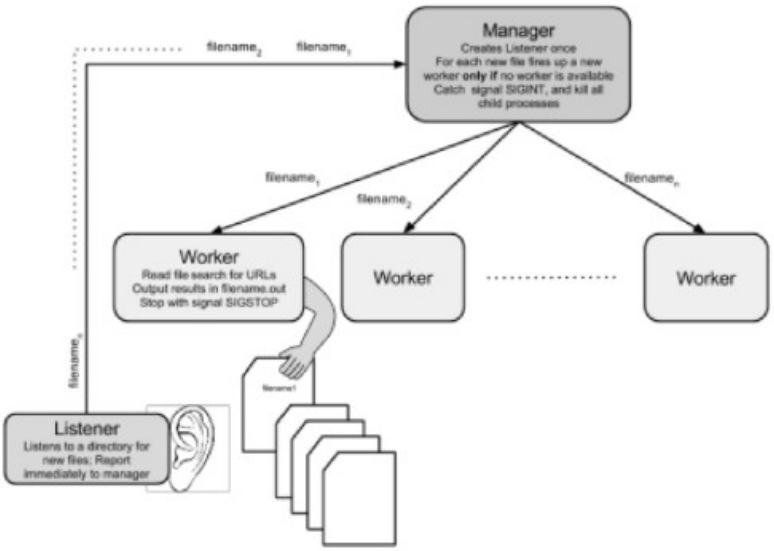
Προγραμματισμός Συστήματος (K24)
Εαρινό εξάμηνο 2021-22

Ονοματεπώνυμο φοιτητη:

- Μαυραπίδης Νικόλαος (Α.Μ.: 11152017 00082)

Αθήνα, 2022

ΣΥΝΤΟΜΗ ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ



Όπως βλέπουμε και στο διάγραμμα, το πρόγραμμα αποτελείται από την διεργασία του **manager**, **listener** και τον/τους **worker/workers**. Η κεντρική διεργασία είναι αυτή του manager, ο οποίος δημιουργεί τον listener και τους workers. Μεταξύ των διεργασιών η επικοινωνία επιτυγχάνεται με την χρήση **pipes**, **named pipes** και **signals**. Οποιαδήποτε διαδικασία τύπου I/O πραγματοποιείται με χρήση συναρτήσων l**ow level I/O** (open, close, write, read etc.). Το πρόγραμμα είναι υλοποιημένο σε γλώσσα **C/C++ και σε περιβάλλον linux**.

ΚΑΤΆΛΟΓΟΣ ΑΡΧΕΊΩΝ ΚΏΔΙΚΑ ΚΑΙ ΕΠΙΚΕΦΑΛΊΔΩΝ

- ***manager.cpp:*** Το αρχείο περιέχει την γονεϊκή **main()** συνάρτηση από όπου δημιουργείται η θυγατρική διαδικασία **listener**. Πιο συγκεκριμένα δημιουργείται μέσω της **fork()** και καλεί μέσω της **exec()** την διαδικασία της **inotifywait** με ορίσματα {**closed_write**, **moved_to**} που κάνει μόνιτορ τον φάκελο που δίνεται από το όρισμα. Έπειτα, αφού ο **listener ειδοποιήσει** για καινούργια αρχεία, ο manager δημιουργεί τους αντίστοιχους **workers** συμβουλεύοντας την ουρά που έχει δημιουργήσει. Σε περίπτωση που βρεθεί στην ουρά worker που είναι available ο manager στέλνει σήμα **SIGCONT** και ξαναχρησιμοποιεί τον ίδιο worker κάνοντας τον resume. Γράφει στους workers τα ονόματα των αρχείων που έχουν έρθει μέσω **named pipes** και περιμένει ξανά για καινούργιο αρχείο από τον listener. Όταν τερματίσει κάποιος worker ο manager λαμβάνει σήμα **SIGCHLD**, βρίσκει ποιο

παιδί σταμάτησε, αλλάζει την διαθεσιμότητά του (availability = true) και τον **ξαναβάζει πίσω στην worker ουρά**. Σε περίπτωση εξόδου του προγράμματος από τον χρήστη με το σήμα **SIGINT**, ο manager τερματίζει τον listener, workers, αποδεσμεύει μνήμη και κλείνει.

- ***worker.cpp:*** Αποτελεί την θυγατρική διεργασία που δημιουργείται μέσω **fork() - execvp() απο την manager** και αναλαμβάνει τα καινούργια αρχεία που έρχοται μέσω listener. Τα ονόματα των αρχείων αυτών μεταφέρονται στους workers μέσω **named pipes** που δημιουργούνται με την σύμβαση ότι κάθε path του pipe αποτελείται από “tmp/fifo” και τέλος τον αριθμό του worker, με την σειρά δημιουργίας τους. Ο κάθε worker αφού λάβει το filename κάνει την διαδικασία που περιγράφεται στην εκφώνηση και **βρίσκει τα URLs**. Αυτό γίνεται με την βοήθεια **regex** και **maps**. Τέλος δημιουργεί αρχεία εξόδου και γράφει τα αποτελέσματα. Κλείνει κάνοντας **raise(SIGSTOP)** στον εαυτό του και περιμένει σε κατάσταση stopped να του ανατεθεί καινούργιο αρχείο. Σε περίπτωση εξόδου και ανάληψης σήματος **SIGTERM** από τον manager, ο worker αποδεσμεύει μνήμη και κλείνει.

- ***aux.cpp:*** Το αρχείο αυτό δημιουργήθηκε καθαρά για βοηθητικούς λόγους και προσφέρει άνεση και ευαναγνωσιμότητα στο κυρίως πρόγραμμα.

- **queueInfo.cpp:** Το αρχείο αυτό δημιουργήθηκε με σκοπό την αποθήκευση πληροφοριών σχετικά με τους workers που υπάρχουν μέσα στην ουρά που διαχειρίζεται ο manager. Συγκεκριμένα δημιουργείται ένα μπλοκ πληροφοριών **{pid, availability, pipe name}** για κάθε worker.

ΠΑΡΑΔΟΧΈΣ

Βασική παραδοχή του προγράμματος είναι ότι όλα τα **αρχεία εξόδου** που παράγονται από το πρόγραμμα (.out άρχεια) μπαίνουν στον φάκελο **./files_output/**

Βασική παραδοχή επίσης είναι ότι ο φάκελος που θα **γίνεται monitor** κάθε φορά από το πρόγραμμα **θα πρέπει να βρίσκεται μέσα στον ευρύτερο φάκελο του προγράμματος**. Για ευκολία έχω συμπεριλάβει στο παραδοτέο έναν φάκελο **./files/** όπου μπορεί ο χρήστης να βάζει εκεί

τα αρχεία. Διαφορετικά μια δημιουργία καινούργιου φακέλου μέσα στο κυρίως directory είναι βασική απαίτηση του προγράμματος.

ΕΚΤΈΛΕΣΗ ΚΑΙ ΧΡΉΣΗ ΠΡΟΓΡΆΜΜΑΤΟΣ

Ο παρεχόμενος φάκελος συμπεριλαμβάνει ένα **makefile** για την μεταγλώττιση όλων των εκτελέσιμων. Τρέχοντας επομένως τις εντολές το πρόγραμμα είναι έτοιμο για χρήση:

make clean και **make**

- Αφότου μεταγλωττιστούν τα αρχεία, ο χρήστης οφείλει να τρέξει την εντολή:

./sniffer [-p path]

Για το δεύτερο όρισμα αρκεί ο user **να κάνει specify μόνο το όνομα** του φακέλου που θέλει να κάνει μόνιτορ. **Επομένως, απαραίτητο είναι ο φάκελος αυτός να βρίσκεται μέσα στον ευρύτερο φάκελο του προγράμματος**. Σε περίπτωση που δεν δοθεί το δεύτερο όρισμα τότε θεωρούμε ως monitored directory **το τρέχον directory**.
Πχ μια **σωστή εκτέλεση του προγράμματος** είναι οι εξής:

./sniffer -p files

ή

./sniffer

- Για την εκτέλεση του **bash script** ο χρήστης πρέπει να τρέξει την εντολή:

./finder.sh [path] [tld_1, tld_2, … tld_n]

Στο όρισμα [path] ο χρήστης **βάζει το directory που περίεχει όλα τα .out αρχεία** ενώ μετά στα υπόλοιπα ορίσματα μπορεί να βάλει ένα ή περισσότερα TLD που θέλει να αναζητήσει.
Πχ μια **σωστή εκτέλεση του script** είναι η εξής:

./finder.sh ./files_output/ gr com gov