

Λειτουργικά Συστήματα

2η υποχρεωτική εργασία

Αναγνώστου Αντώνης – 2268

Γωγούσης Παύλος – 2251

Λασκαρίδης Στέφανος - 2315

Τμήμα Πληροφορικής ΑΠΘ
2-13-2015

Περιεχόμενα

Περιγραφή εργασίας	2
Γενικές παραδοχές	2
Βασικές οντότητες	2
Βασικές κλάσεις	2
Scheduler	2
Command	2
CommandPrompt	2
Utils	3
Σημείο έναρξης (main)	3
Προβλήματα.....	3
Μελλοντικές επεκτάσεις	4
Σύνοψη.....	5
Βιβλιογραφία.....	6

Η παρούσα εργασία αποτελεί την δεύτερη υποχρεωτική άσκηση για το μάθημα των Λειτουργικών Συστημάτων και αφορά την κατασκευή ενός κελύφους (shell) για το GNU Linux.

Για την ανάπτυξη χρησιμοποιήθηκε το περιβάλλον Eclipse Luna (for C++), C++1y σε λειτουργικό σύστημα Ubuntu 12.04 και 14.04 LTS x64.

Στην τρέχουσα αναφορά μπορείτε να βρείτε τις γενικές παραδοχές και τα προβλήματα που αντιμετωπίσαμε κατά την ανάπτυξη του προγράμματος, ενώ λεπτομερή περιγραφή των μεθόδων και κλήσεων που χρησιμοποιήθηκαν βρίσκονται στον πηγαίο κώδικα του προγράμματος.

Γενικές παραδοχές

Οι παραδοχές που έγιναν κατά την ανάπτυξη του παραδοτέου κώδικα είναι οι ακόλουθες:

- Δεν υποστηρίζεται η εκτέλεση εντολών με wildcard χαρακτήρα (πχ. '*')
- Δεν υποστηρίζεται η εκτέλεση διασωληνούμενων εντολών, εκ των οποίων η δεύτερη τρέχει στο παρασκήνιο.

Βασικές οντότητες

Βασικές κλάσεις

Οι βασικές κλάσεις που χρησιμοποιήθηκαν είναι οι εξής:

Scheduler

Πρόκειται για τη βασική κλάση του δρομολογήτη. Κρατά εσωτερικά τα pids των διεργασιών που του έχουν υποβληθεί, ενώ εκκινείται ανά 1 sec, πραγματοποιώντας μία Round Robin δρομολόγηση σ'αυτά.

Command

Πρόκειται για τη βασική κλάση που περιγράφει μία «εντολή», δηλ. μία είσοδο του χρήστη. Περιέχει το όνομα του εκτελέσιμου προγράμματος, τις παραμέτρους αυτού, πιθανούς τελεστές, κλπ.

CommandPrompt

Πρόκειται για την κλάση που περιγράφει το interface του shell με το χρήστη. Δοθείσας μίας συμβολοσειράς εισόδου, δημιουργεί αντικείμενα από την κλάση Command, αναγνωρίζοντας επιμέρους σημεία της (πχ. ανακατεύθυνση, pipelining, κλπ).

Utils

Πρόκειται για μια βοηθητική κλάση, που περιέχει στατικές μεθόδους και συναρτήσεις που χρησιμοποιούνται από όλα τα σημεία του προγράμματος. Για παράδειγμα, μια τέτοια μέθοδος είναι η `tokenize`, που έχει ευρεία χρήση, καθώς χρησιμοποιείται με οποιονδήποτε `delimiter`.

Σημείο έναρξης (`main`)

Κατά την εκτέλεση του προγράμματος, συμβαίνουν διαδοχικά τα ακόλουθα:

- Πραγματοποιείται `fork`, ώστε να αρχικοποιηθεί η διεργασία του `Scheduler`.
- Η κύρια διεργασία εκτέλεσης αποτελεί τον `parser (CommandPrompt)` του τερματικού.
- Σε περίπτωση εκτέλεσης στο υπόβαθρο, οι διεργασίες υποβάλλονται στον `Scheduler`, διαφορετικά αποτελούν παιδιά της κύριας διεργασίας του `CommandPrompt`.

Προβλήματα

Κατά την διάρκεια της υλοποίησης, αντιμετωπίσαμε τα παρακάτω προβλήματα:

- Υλοποίηση μίας αποδοτικής συνάρτησης αποκοπής συμβολοσειρών και διαχωρισμού των λέξεων με βάση έναν χαρακτήρα (`tokenize`). Αν και υπάρχουν ήδη σχετικές υλοποιήσεις, καμία δεν κάλυψε τις ανάγκες της παρούσας εργασίας, γι' αυτό και οδηγηθήκαμε στην ανάπτυξη μίας εκ νέου.
- Επικοινωνία του δρομολογητή των εντολών στο παρασκήνιο με το κύριο πρόγραμμα, προκειμένου να χειρίζεται τις διεργασίες αυτές (αποστολή σημάτων `SIGCONT`, `SIGSTOP`, με αλγόριθμο `Round Robin`). Αρχικά, η παρακάτω λειτουργία υλοποιήθηκε με τη χρήση αρχείων στα οποία κρατώνταν τα `pids` των διεργασιών στο παρασκήνιο. Ωστόσο, η υλοποίηση αυτή σύντομα οδήγησε σε προβλήματα. Πρώτα, ο δρομολογητής δεν μπορούσε να στείλει σήμα `SIGSTOP` στις διεργασίες αυτές, καθώς δεν ήταν `child processes` του. Έπειτα, δημιουργούνταν θέματα ανταγωνισμού (`race conditions`) για την ανάγνωση/εγγραφή των

pids από την διεργασία του CommandPrompt και του Scheduler αντίστοιχα, λόγω ενδεχόμενης χρονικής σύμπτωσης. Το συγκεκριμένο πρόβλημα, επιλύθηκε με την εισαγωγή διπλής διασωλήνωσης μεταξύ των προαναφερθέντων διεργασιών. Συγκεκριμένα, το ένα ζεύγος περιγραφών αρχείων αφορά στην ανταλλαγή αντικειμένων τύπου «Command», ενώ το δεύτερο στην παρουσία μίας μεταβλητής που δηλώνει το πλήθος των εντολών που υπάρχουν στον περιγραφέα αρχείων για λόγους συγχρονισμού. Κατ'αυτό τον τρόπο, ο δρομολογητής «διαβάζει» το πλήθος των εντολών που πρόκειται να συναντήσει στον περιγραφέα, και αν αυτό είναι θετικό, τότε και μόνο τότε προχωρά στην ανάγνωση αυτών.

- Ο τερματισμός του προγράμματος θα πρέπει να συνεπάγεται τερματισμό όλων των διεργασιών και πόρων που έχουν δεσμευτεί από το κέλυφος. Εφ'όσον ο δρομολογητής βρίσκεται σε διαφορετική διεργασία από την γραμμή εντολών, βασίσαμε την λειτουργικότητα αυτή στην χρήση ενός signal handler (σήμα τύπου SIGINT) από τον δρομολογητή. Όταν ο τελευταίος λάβει το σήμα αυτό, τότε σκοτώνει (σήμα SIGKILL) όλες τις διεργασίες στο παρασκήνιο, απελευθερώνει τυχόν πόρους, και τερματίζει.

Μελλοντικές επεκτάσεις

Στις μελλοντικές επεκτάσεις του προγράμματος, θα μπορούσαν να συμπεριληφθούν τα ακόλουθα:

- υποστήριξη διασωλήνωσης στο παρασκήνιο με χρήση command grouping
- πολλαπλή διασωλήνωση
- χρήση τελεστή * (wildcard)
- αυτόματη συμπλήρωση ονομάτων εντολών και αρχείων.
- αλλαγή των ονομάτων των forked processes

Σύνοψη

Το κέλυφος που υλοποιήθηκε ζητά εντολές από τον χρήστη έως ότου πληκτρολογηθεί exit. Κατόπιν, γίνεται parsing και τεμαχισμός της εντολής με την βοήθεια της συνάρτησης tokenize. Αφού αναγνωρισθεί η εντολή εκτελείται το ανάλογο τμήμα κώδικα. Αναφορικά, το κέλυφος υποστηρίζει τις ακόλουθες ενέργειες:

1. Αλλαγή του τρέχοντος καταλόγου σε εναλλακτικό δοθέντα κατάλογο – εντολή cd.
2. Ανακατεύθυνση της εισόδου ή εξόδου μιας εντολής από ή προς ένα αρχείο – εντολές "<", ">", ">>" με ανάλογα ορίσματα.
3. Διασωλήνωση μεταξύ δύο διεργασιών μέσω του τελεστή "|".
4. Εκτέλεση διεργασιών στο υπόβαθρο – τελεστής "&" στο τέλος της εντολής
5. Οι διεργασίες που βρίσκονται στο υπόβαθρο δρομολογούνται σύμφωνα με τον αλγόριθμο RR (Round Robin) - χρονοπρογραμματισμός εξυπηρέτησης εκ περιτροπής.

Βιβλιογραφία

- Ανάκτηση από Stack Overflow:
<http://stackoverflow.com/questions/395877/are-child-processes-created-with-fork-automatically-killed-when-the-parent-is>
- Ανάκτηση από Stack Overflow:
<http://stackoverflow.com/questions/16078985/why-zombie-processes-exist>
- Briggs, G. Ανάκτηση από University of Rochester alumni:
<http://www.uofr.net/~greg/processname.html>
- C++ Reference Website. Ανάκτηση από <http://www.cplusplus.com/>
- *die.net*. Ανάκτηση από <http://www.die.net>
- N. Passalis, N. K. Pileas. Ανάκτηση από
<http://pileas.csd.auth.gr/course/view.php?id=149>