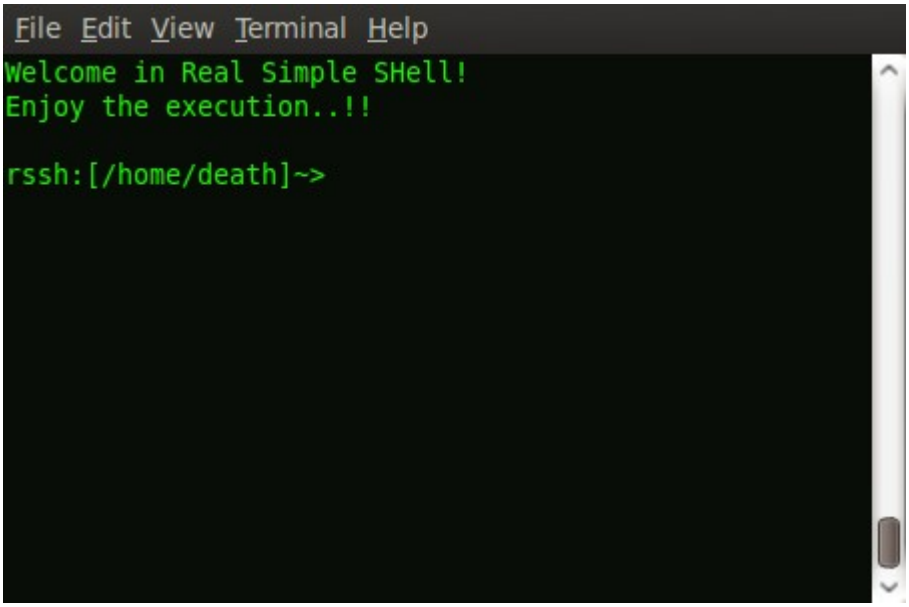
Real Simple Shell



Ονοματεπώνυμο Α.Μ.	Ψαλλίδας Φώτης 1115200600170
Μάθημα	Προγραμματισμός Συστήματος
Αντικείμενο Εργασίας	Υλοποίηση Shell
Καθηγητής	κ. Αλέξης Δελλής
Ακαδημαικό Έτος	2008-2009

Αρχεία

- rssh.hpp
- rssh.cpp
- wild.hpp
- path.hpp
- history
- rsshrc
- Makefile

Επεξήγηση-Λειτουργικότητα Αρχείων

rssh.hpp

Header file όπου υλοποιείται το core του rssh.
Πιο συγκεκριμένα ορίζονται οι εξής κλάσσεις

class Alias

Υλοποίηση συναρτήσεων για την διαχείρηση των aliases
Καθε alias αποτελείται από το from και από το to
Στο from αποθηκεύεται το string που αποτελεί το alias
ενώ στο to αποθηκεύεται η εντολή που θα αποκαταστήσει
το alias.

class Var

Υλοποίση συναρτησεων για την διαχείρηση των variables.
Καθε variable αποτελείται από ένα name και ένα value.
Σημαντικές μεταβλητές για το rssh ειναι οι

- PATH : Directories που θα ψάξει το rssh για εκτελέσιμα,
- HISTORY : Μέγεθος πίνακα ιστορικού(by deafult 10)
- HOME : Directory από όπου ξεκινάει η εκτέλεση των εντολών

class rssh

Βασική κλάσση υλοποίησης όλως της λειτουργικότητας του shell.
Υποκλάσσεις:

- Command : Κλασση που κρατάει τα δεδομένα για εντολές που δίνει ο χρήστης.
- Job: Κλάσση εφάμιλλη της Command με την διαφορ οτι χρησιμοποείται μόνο στην περίπτωση που έχει γίνει define το TTY(βλ.TTY).
- History: Κλάσση διαχείρησης του history board
  Σε πρώτη φάση αν ο χρήστης έχει δώσει HISTORY size μέσω του rsshrc γινεται alloc ο πίνακας που κρατάει το ιστορικό εντολών.Αν υπάρχει .history τοτε γίνονται append στο πίνακα μόνο history size jobs.Στο τέλος(quit,exit,q) το history table γράφεται στο αρχείο .history.Για κάθε μια εντολή το history κρατάει το χρόνο που εκτελέστηκε η job ,την ίδια την εντολή που εκτελέστηκε,καθώς και ένα History ID (Hid) με το οποίο ο χρήστης μπορεί να επανεκτελέσει καλέσει κάποια συγκεκριμένη εντολή που έτρεξε στο παρελθόν.Σε περίπτωση που ο χρήστης δώσει εντολή setenv $HISTORY = size ,τοτε το HISTORY γινεται resize και κρατάει τις size πιο πρόσφατες εντολές.Τελος στο history table δεν ανεβαίνουν οι inline εντολές ,showhistory και history ,εντολες που θα επεξεγηθούν αργότερα.

Data που κρατάει η rssh

- lalias: Λίστα με όλα τα aliases
- lvar: Λίστα με όλες τις μεταβλητές
- history : Ενας pointer σε History object για να κρατάει το ιστορικό εντολών.
- rcd: Τον κατάλογο που βρίσκεται το rsshrc
- cwd: Τρέχων κατάλογος εργασίας
- home: Ο home κατάλογος του χρήστη
- PATH: Ολοι οι κατάλογοι που βρίσκονται εκτελέσιμα αρχεία
  Οι διάφοροι κατάλογοι χωρίζονται με ':' ενώ στο PATH ενσωματώνεται αυτόματα και ο cwd.
- newmask,oldmask: Μασκες για τα signals που δέχεται το rssh.
- curpid: Pid της job που τρέχει στο foreground.Χρησιμεύει για τις περιπτώσεις που στέλνει ο χρήστης Ctrl-c έτσι ώστε να αποσταλει SIGKILL στην διεργασία που τρέχει στο foreground.Αν δεν τρέχει κάποια διεργασία τότε το curpid ειναι -1.

Τα υπόλοιπα δεδομένα που κρατάει το rssh χρησιμοποιούνται για τη διαχείρηση του tty του terminal..

Λειτουργίες της rssh:

1. Ανάλυση εντολών(λεκτική συντακτική γραμμάτική) και εκτέλεση γράφου εντολών:

- **remove_white(std::string)**
  Διαγραφή των κενών από την αρχή και το τέλος ενός string.
- **void parser()**
  Ρουτίνα που υλοποεί το interactive κομμάτι του rssh με το χρηστη.
  Ο χρήστης δίνει μια εντολή-γραμμή και ο parser ειναι υπεύθυνος να πάρει αυτη τη γραμμή να την αναλύσει και να την εκτελέσει χρησιμοποιώντας την ρουτίνα command_parser..
- **void command_parser(std::string)**
  Στην ρουτίνα αυτή γίνεται η αναλυση της job,γίνεται ο κατάλληλος έλεγχος της κάθε διεργασίας (check),και διαμορφώνεται ο τελικός γράφος προς εκτέλεση.Αυτό γινεται με την χρήση των ρουτινών
  check(έλεγχος)
  make_commands(δημιουργία τελικού γράφου)
  exec_checking(ελεγχος για executable files καθώς και διαμόρφωση των arguments των εντολών)
- **bool check(PList<std::string>* , Plist<std::string>*)**
  Έλεγχος των λεκτικών και συντακτικών κανόνων
  Υπάρχει αναλυτική επεξήγηση των κανόνων αυτών μέσα στο rssh.hpp.
- **std::string* manage_wild_args(std::string*,bool)**
  Ρουτίνα για matching μεταξύ patterns και αρχείων μέσα στο τρέχοντα κατάλογο.
- **std::string fix(std::string)**
  Ρουτίνα που κάνει split τους operators < > ; & | >> από τις λέξεις
  Επισης σε περιπτώσεις με πολλά ';' διαγράφονται και παραμένει μόνο ενα.
- **template<class S>**
  int exec_checking(PList<S>*,PList<std::string>* pplist)
  Αποσαφήνιση των arguments των διαφόρων εντολών.
  Εύρεση του path για το executable της εντολής κάτω από το PATH.
- **template<class S>**
  PList<S>* make_commands(PList<std::string>*pplist,PList<std::string>*)
  Δημιουργία ημιτελή γράφου για εκτέλεση.Στον γράφος αυτό απλά έχουν διαχωριστεί τα arguments με τις εντολές από τους operators.
  Επίσης έχουν αποσαφηνιστεί τα input και output της κάθε εντολής.
- **template<class S>**
  int run(PList<S> **,int* ,int,bool*)
  Η ρουτινα αυτή ειναι υπεύθυνη για την εκτέλεση του τελικού γράφου εντολών
  Δημιουργεί αρχικά ένα dummy shell του αρχικού shell.Αν η job ειναι background τότε δημιουργείται και ενα άλλο dummy shell κάτω από το υπαρχων dummy shell ωστε να μην μεινουν zombie οι εντολές αλλά orphanes και να τις διαχειριστεί η init.
  Αν δεν ειναι background η εντολή το shell περιμένει να τελειώσει και όταν τελειώσει επανακτά το tty του terminal και συνεχίζει με τις υπόλοιπες εντολές.
  Αν δώσουμε background εντολή-εντολές τότε αυτες θα τρέξουν concurrently
  ls & ; ls & ; ls
  Ολες οι παραπάνω εντολές θα τρέξουν ταυτόχρονα ενώ το shell θα περιμένει

να τελειώσει η τελευταία ls.
Στην περίπτωση που δωθεί
ls & ; ls & ; ls ; cat file1
Οι εντολές ls & ; ls & ; ls ; θα τρεξουν ταυτόχρονα ενώ η εντολή cat file1 θα περιμένει να τελειώσει η εντολή ls ;.

2. Ρουτίνες για τις inline εντολές

- **int create_alias(std::string,unsigned int)**
  Εισαγωγή alias στη λίστα με τα alias(lalias)
- **int destroy_alias(std::string)**
  Διαγραφή alias από τη λίστα με τα alias(lalias)
- **int setenv(std::string)**
  Εισαγωγή μεταβλητών στη λίστα με τις μεταβλητές
- **int cd(std::string)**
  Μεταφορά του ελέγχου σε συγκεκριμένο κατάλογο.
- **void show_history()**
  Εκτύπωση του history table.Χρησιμο για να πάιρνουμε το hid.
- **template<class S>**
  int inline_cmds(PList<S>*)
  Ανάλυση αν μια συγκεκριμένη εντολή ειναι inline και αντιστοιχη κλήση της κατάλληλης ρουτίνας υλοποίησης της inline εντολής.

3. Ρουτίνες διαχείρησης αρχείων

- **void readrc()**
  Ρουτίνα διαβάσματος του rsshrc.Μέσα στο rsshrc υπάρχουν aliases και variables.
- **void readhis()**
  Ρουτίνα διαβάσματος του history.
- **void writehis()**
  Ρουτίνα για γράψιμο του history table στο history.
- **void writerc()**
  Ρουτίνα για γράψιμο των aliases και των variables στο rsshrc

Δομή rsshrc

```
%        :    comment line
alias ..  :    alias line
other .. :    variable
```

Δομή history:

```
time
job

time
job
....
```

4. Αρχικοποίηση μεταβλητών και τερματικού

- **void initialize()**
  Καθαρισμός οθόνης ,μεταφορά ελέγχου στο home directory
- **void init_shell ()**
  Αποθήκευση των δεδομένων του tty του terminal.

rssh.cpp

Αρχείο από όπου ξεκινάει η εκτέλεση του rssh. Σε πρώτη φάση διβάζονται τα αρχεία history και rsshrc ενώ μετά από αυτά ξεκινάει ο parser με το interactive κομματι και περιμένει από το χρήστη την είσοδο του.
Αν γίνει exit του rssh τοτε καλειται η clean η οποία διαγράφει τη μνήμη και επίσης γράφει τα aliases και variables στο rsshrc και το history table στο history.

wild.hpp

Υλοποιεί το pattern matching μεταξύ δυο string.Ελέγχει τις περιπτωσεις {},*,?,~,+,[]
Μέσα στα {} αν δωθει σύμβολα pattern θα χρησιμοποιηθεί ως απλός character.
Χρησιμοποιειται απο την συνάρτηση manage_wild_args του rssh.hpp
έτσι ώστε να βρεθουν τα αρχεια που κανουν match με κάποιο δωθεν pattern.

path.hpp

Υλοποιήση συνάρτησης fix_path η οποία δοθεντος του τρέχοντα καταλόγου
και του σχετικού path ενός file επιστρέφει το απόλυτο μονοπάτι.
Χρησιμοποείται στην περίπτωση της cd αλλά και στην περίπτωση
που ψάχνουμε ενα executable κάτω από το path.Στην περίπτωση
που βρούμε το αρχείο που ψάχναμε αλλά ειναι link η συνάρτηση αυτή χρησιμοποείται για να μας επιστρέψει το path του linked αρχείου.

history

Αρχείο με το history όλων των εντολών που τρέξαμε στο παρελθον (την προηγούμενη φορά που τρέξαμε το rssh). Καθε φορά που τρέχουμε το rssh θα διαβαστουν οι προηγούμενες εντολές που τρέξαμε και θα ανεβούν στο history table.
Πρεπει να βρίσκεται στο ίδιο directory που βρίσκεται και το rssh.

rsshrc

Αρχειο που γράφονται τα variables και τα aliases.Καθε φορά που τρέχουμε το rssh θα διαβαστουν τα aliases και τα variables από το rsshrc και θα ανεβούν σε κατάλληλες δομές.
Στο τέλος θα γραφτούν και πάλι τα aliases και τα variables στο αρχείο rsshrc.
Πρεπει να βρίσκεται στο ίδιο directory που βρίσκεται και το rssh.

Makefile

Χρησιμοποείται για αυτοματη μεταγλώτιση.
Πατήστε απλά make.Τότε θα δημιουργήθει στο τρέχοντα κατάλογο το rssh.
Τρέξτε το rssh.
Αν κάνετε compile σε sun του τμήματος κάντε comment out την σειρά που αναγράφεται μέσα στο makefile και έπειτα τρέξτε το make.

Περαιτέρω σχολιασμός του rssh μπορείτε να βρείτε inline μέσα στα παραπάνω αρχεία..