

Tehtävä 2
Petteri Mäkelä
16/12/18

Dokumentaatio ja ajoja

Harjoitustyön tehtävänä oli tehdä shell. Shell toimii halutulla tavalla, ja se ei ole erittäin virhealtis. Testausta eri virheille olisi voinut tehdä enemmän. Shelliä olisi voinut myös testata eri testausohjelmilla esimerkiksi muistivuotojen varalta. Lopputulos on kuitenkin hyvä, shelliä käyttäessä pitää tavoitteena olla sen rikkominen, mikäli jotain haluaa pieleen menevän.

Shell koostuu pääohjelmasta, sekä aliohjelmista. Pääohjelmassa käsitellään annettu rivi, sekä toteutetaan while -loopin sisällä toimivat eri komentojen toiminnallisuudet. Pääohjelman alussa määritellään käytetyt kirjastot, funktiot, sekä muuttujat. Suurin osa muuttujista on staattisia, joka voi aiheuttaa ongelmia kun shelliä ajetaan äärimmäisyyksissä.

Shellin toiminnoissa ovat mukana funktiot `get_command`, `check_command`, sekä `run_command`. Tämän lisäksi komennoille, jotka eivät ole shelliin sisäänrakennettuja, suoritetaan `check_flags` testaus redirectoinnin (`>`) varalta.

`get_command` Käsittelee käyttäjän syötteen. Tämän jälkeen syöte parsitaan haluttuun muotoon pääohjelman sisällä.

`check_command` tarkistaa komennon tyyppin. Se ohjaa pääohjelman `switch():n` kautta oikeaan kohtaan. 0 = exit, 1 = komento kirjastosta, 2... = built-in komennot. -1 palautetaan, mikäli komentoa ei löydy, tai muita virheitä tapahtui.

Exit -komennon suorittaminen oli helppoa implementoida. Se tehdään kokonaan pääohjelman puolella, ja se kutsuu `exit()` komentoa, sekä tulostaa käyttäjälle kiitos viestin. Pääohjelman lopussa on `return(-1)` rivi, joka palauttaa ohjelmasta -1, mikäli tänne ollaan päästy. Pääohjelman loppuun ei tulisi ikinä päästä, sillä ohjelmasta poistutaan tappamalla se, tai exit komennolla.

`my_cd` toimii halutulla tavalla. Se pystyy siirtymään sekä tiettyyn kansioon, mikäli syötteen ensimmäinen kirjain on `'/'`. Mikäli syöte on `'..'`, siirrytään yksi kansiotason alaspäin. Muilla syötteillä pyritään siirtymään ylöspäin kansiorakenteessa.

`my_path` toimii myös halutulla tavalla. Sitä kutsuessa `commandpath`, eli komentojen sisältämät kirjastot nollaantuvat. Tämän jälkeen uusiksi kirjastoiksi asetetaan ylimääräiset flagit. Esimerkiksi näin: `path /bin/ /usr/bin/`.

Komentojen suorittaminen toimii hyvin. Niiden output voidaan halutessa ohjata tiedostoon (`>`) Tiedoston myös luodaan ohjelmaa suorittaessa. Run command perustuu kutsuun `fork()`, joka luo pääohjelmasta, eli shellistä, kopioidun lapsi-ohjelman. Pääohjelma odottaa, jotta lapsi on suoritettu, jonka jälkeen se jatkaa suorittamistaan, eli shell jatkaa pyörimistään.

Ohjelmien rinnakkaisajoa ei ole suoritettu. Päästässeni tähän vaiheeseen en nähnyt mahdollisuutta toteuttaa sitä tekemälläni pohjalla. Se olisi mahdollista tekemällä uusi `run_command` -tyylinen funktio, mutta siihenkin kuluisi enemmän aikaa kuin minulla on antaa.

Virheentarkistuksissa on käytetty haluttua muotoa. Virheen tarkistus on suoritettu malloc(), getcwd(), my_cd(), sekä check_command() funktioille. Nämä funktiot palauttavat tapahtuneen virheen, sekä virheviestin, mikäli virhe on tapahtunut. Virheentarkastus voisi kuitenkin olla parempaa, ja ohjelmasta varmasti löytyy muitakin virheitä sitä tarpeeksi testaamalla.

Alla on kopioituna ohjelman eri ajoja. Ensin on ohjelman kääntäminen, ja sen päänäkymä. Toisena on batch moden testaaminen tavallisilla komennoilla. Tämän jälkeen ohjelmaa testataan kokonaisuudessaan input modessa, käyden läpi mahdollisimman huolellisesti sen kaikki toiminnallisuudet.

```
petteri@petteri-Latitude-3330:~$ cd Documents/ktt/ht2
petteri@petteri-Latitude-3330:~/Documents/ktt/ht2$ ls
asd.txt batch.txt new2.txt new.txt wish wish.c
petteri@petteri-Latitude-3330:~/Documents/ktt/ht2$ gcc wish.c -o wish -Wall -Werror
petteri@petteri-Latitude-3330:~/Documents/ktt/ht2$ ./wish
Welcome to wish shell!
Working directory: /home/petteri/Documents/ktt/ht2
Commandpath 1: /bin/
Type a command!
```

Batch modella:

petteri@petteri-Latitude-3330:~/Documents/ktt/ht2\$./wish batch.txt

Welcome to wish shell!

Working directory: /home/petteri/Documents/ktt/ht2

Commandpath 1: /bin/

Command given: ls

Executing command: ls

asd.txt batch.txt new2.txt new.txt wish wish.c

.....

Working directory: /home/petteri/Documents/ktt/ht2

Commandpath 1: /bin/

Command given: echo "asd"

Executing command: echo

"asd"

.....

Working directory: /home/petteri/Documents/ktt/ht2

Commandpath 1: /bin/

Command given: cat batch.txt

Executing command: cat

ls

echo "asd"

cat batch.txt

grep main wish.c

.....

Working directory: /home/petteri/Documents/ktt/ht2

Commandpath 1: /bin/

Command given: grep main wish.c

Executing command: grep

int main(int argc, char **argv) {

.....

Working directory: /home/petteri/Documents/ktt/ht2

Commandpath 1: /bin/

Welcome to wish shell!
Working directory: /home/petteri/Documents/ktt/ht2
Commandpath 1: /bin/
Type a command!
path
Command given: path

.....
Working directory: /home/petteri/Documents/ktt/ht2
Type a command!
ls
Command given: ls
Command not found
An error has occurred

.....
Working directory: /home/petteri/Documents/ktt/ht2
Type a command!
path /bin/ /bin/user/
Command given: path /bin/ /bin/user/

.....
Working directory: /home/petteri/Documents/ktt/ht2
Commandpath 1: /bin/
Commandpath 2: /bin/user/
Type a command!
ls
Command given: ls
Executing command: ls

asd.txt batch.txt new2.txt new.txt wish wish.c

.....
Working directory: /home/petteri/Documents/ktt/ht2
Commandpath 1: /bin/
Commandpath 2: /bin/user/
Type a command!
cd ..
Command given: cd ..

.....
Working directory: /home/petteri/Documents/ktt
Commandpath 1: /bin/
Commandpath 2: /bin/user/
Type a command!
cd ht2
Command given: cd ht2

.....
Working directory: /home/petteri/Documents/ktt/ht2
Commandpath 1: /bin/
Commandpath 2: /bin/user/

Type a command!

cd /

Command given: cd /

.....

Working directory: /

Commandpath 1: /bin/

Commandpath 2: /bin/user/

Type a command!

cd /

Command given: cd /

.....

Working directory: /

Commandpath 1: /bin/

Type a command!

cd /home/petteri/Documents/ktt/ht2

Command given: cd /home/petteri/Documents/ktt/ht2

.....

Working directory: /home/petteri/Documents/ktt/ht2

Commandpath 1: /bin/

Type a command!

ls / > resultfile.txt

Command given: ls / > resultfile.txt

Executing command: ls

.....

Working directory: /home/petteri/Documents/ktt/ht2

Commandpath 1: /bin/

Type a command!

cat resultfile.txt

Command given: cat resultfile.txt

Executing command: cat

bin

boot

cdrom

dev

etc

home

initrd.img

initrd.img.old

lib

lib32

lib64

lost+found

media

mnt

opt

proc

root
run
sbin
snap
srv
swapfile
sys
tmp
usr
var
vmlinuz
vmlinuz.old

.....

Working directory: /home/petteri/Documents/ktt/ht2

Commandpath 1: /bin/

Type a command!

exit

Command given: exit

Bye Bye