

Käyttöjärjestelmät ja systeemiohjelmointi

Olga Vinogradova-Lebedeva
System Programming

16. kesäkuuta 2024

1 Project 3

1.1 Yleiskatsaus

Ohjelma toimii komentotulkina, joka pystyy suorittamaan komentoja joko interaktiivisessa tilassa tai batch-tilassa, käsittelemään useita komentoja rinnakkain sekä ohjaamaan tulosteet tiedostoon.

1.2 Tiedostot ja Funktiot

1.2.1 Määritelmät ja vakio-arvot

- `#define MAX_ARGS 100`: Maksimimäärä argumentteja komentoa kohden.
- `#define MAX_PATHS 100`: Maksimimäärä polkuja hakemistoja varten.
- `#define BUFFER_SIZE 1024`: Puskurin koko tiedoston lukemista varten.
- `char error_message[30] = "An error has occurred"`: Virheviesti, joka tulostetaan virhetilanteessa.
- `char *paths[MAX_PATHS]`: Taulukko, joka sisältää polut suoritettaviin komentoihin.
- `int num_paths = 1`: Polkujen määrä, alustettu arvoon 1.

1.2.2 Funktiot

`execute_command`

Suorittaa annetun komennon määritetyissä poluissa.

Parametrit:

- `char **args`: Komennon ja sen argumenttien taulukko.

1.2.3 execute_parallel_commands

Tarkoitus: Suorittaa useita komentoja rinnakkain.

Parametrit:

- `char *line`: Rivi, joka sisältää useita komentoja eroteltuina &-merkillä.

process_input

Tarkoitus: Käsitellä annettu komento tai komennot, mukaan lukien sisäänrakennetut komennot ja uudelleenohjaukset.

Parametrit:

- `char *line`: Käsiteltävä komento.

set_path

Asettaa polut, joissa komentoja etsitään.

Parametrit:

- `char **args`: Taulukko, joka sisältää polut.

change_directory

Vaihtaa nykyinen työhakemisto.

Parametrit:

- `char **args`: Taulukko, joka sisältää komennon ja hakemiston.

handle_redirection

Käsitellä komentojen uudelleenohjaukset.

Parametrit:

- `char **args`: Taulukko, joka sisältää komennon ja sen argumentit.

error

Tulostaa virheviesti standard-virhevirtaan.

trim_whitespace

Poistaa ylimääräiset välilyönnit annetusta merkkijonosta.

Parametrit:

- `char *str`: Merkkijono, josta poistetaan välilyönnit.

1.3 Pääohjelma (main)

Tarkistaa komentorivin argumentit ja suorittaa ohjelman joko interaktiivisessa tai batch-tilassa.

Parametrit:

- `int argc`: Argumenttien määrä.
- `char *argv[]`: Argumenttivektori.

Toiminta:

- Jos argumentteja on alle kaksi, ohjelma toimii interaktiivisessa tilassa.
- Jos argumentteja on tasan kaksi, ohjelma toimii batch-tilassa ja lukee komennot tiedostosta.
- Jos argumentteja on enemmän kuin kaksi, ohjelma tulostaa virheviestin ja lopettaa virhekoodilla 1.

1.4 Kääntäminen ja Suorittaminen

Koodin kääntäminen:

```
gcc -o wish my-shell.c -Wall -Werror
```

Ohjelman suorittaminen interaktiivisessa tilassa:

```
./wish
```

```
+ systeemiohjelmointi git:(project-3) gcc -o my-shell my-shell.c -Wall -Werror
+ systeemiohjelmointi git:(project-3) x ./my-shell
wish> ls -la .
total 576
drwxr-xr-x 20 clara staff 640 Jun 15 23:09 .
drwxr-xr-x 5 clara staff 160 Jun 15 22:28 ..
-rw-r--r--@ 1 clara staff 6148 Jun 15 23:08 .DS_Store
drwxr-xr-x 14 clara staff 448 Jun 15 23:08 .git
-rw-r--r-- 1 clara staff 9 Jun 15 22:24 .gitignore
-rw-r--r-- 1 clara staff 25 Jun 15 23:03 file.z
-rw-r--r-- 1 clara staff 7410 Jun 15 22:21 main.c
-rw-r--r-- 1 clara staff 1187 Jun 15 23:03 my-cat.c
-rw-r--r-- 1 clara staff 1417 Jun 15 23:03 my-grep.c
-rwxr-xr-x 1 clara staff 51624 Jun 15 23:09 my-shell
-rw-r--r-- 1 clara staff 8010 Jun 15 23:07 my-shell.c
-rw-r--r-- 1 clara staff 1091 Jun 15 23:03 my-unzip.c
-rw-r--r-- 1 clara staff 1726 Jun 15 23:03 my-zip.c
-rwxr-xr-x 1 clara staff 34344 Jun 15 22:26 reverse
-rw-r--r-- 1 clara staff 14 Jun 15 22:30 sample_input.txt
-rw-r--r-- 1 clara staff 13 Jun 15 23:03 sample_output.txt
-rwxr-xr-x 1 clara staff 33736 Jun 15 23:03 wcac
-rwxr-xr-x 1 clara staff 33832 Jun 15 23:03 wgrep
-rwxr-xr-x 1 clara staff 33784 Jun 15 23:03 wunzip
-rwxr-xr-x 1 clara staff 33784 Jun 15 23:03 wzip
wish> exit
+ systeemiohjelmointi git:(project-3) x █
```

Kuva 1: My-shell käyttö

Ohjelman suorittaminen batch-tilassa:

```
./wish batchfile
```

1.5 Virheen käsittely

Ohjelma käsittelee virheitä tiedostojen avaamisessa, komentojen suorituksessa ja uudelleenohjauksessa. Jos tiedoston avaaminen epäonnistuu, se tulostaa virheilmoituksen standard-virhevirtaan ja lopettaa tilakoodilla 1.

1.6 Testit

Ohjelma on läpäissyt melkein kaikki Ostep-projektin testit (3) paitsi 11 testi. Siellä argumentit ovat annettu ilman välilyöntiä ja ohjelma ei pysty siihen. Tässä se on muutettu, ja muut testit on läpäisty.

Ohjelmaa kutsutaan **wish** testien läpäisemiseen.

```
diff --git a/processes-shell/tests/11.in b/processes-shell/tests/11.in
index 1948c9f..4bb0cf9 100644
--- a/processes-shell/tests/11.in
+++ b/processes-shell/tests/11.in
@@ -1,4 +1,4 @@
-ls tests/p2a-test>/tmp/output11
+ls tests/p2a-test > /tmp/output11
cat /tmp/output11
rm -f /tmp/output11
exit
(END)
```

Kuva 2: My-shell 11.testin muutos

```
→ processes-shell git:(master) ✖ cp ../../systemiohjelmointi/my-shell wish
→ processes-shell git:(master) ✖ sh test-wish.sh
test 1: passed
test 2: passed
test 3: passed
test 4: passed
test 5: passed
test 6: passed
test 7: passed
test 8: passed
test 9: passed
test 10: passed
test 11: passed
test 12: passed
test 13: passed
test 14: passed
test 15: passed
test 16: passed
test 17: passed
test 18: passed
test 19: passed
test 20: passed
test 21: passed
test 22: passed
→ processes-shell git:(master) ✖
```

Kuva 3: My-shell testit