

Käyttöjärjestelmät ja systeemiohjelmointi

Olga Vinogradova-Lebedeva
System Programming

16. kesäkuuta 2024

1 Project 2

Tässä projektissa toteutetaan 4 ohjelmaa: my-cat, my-grep, my-zip ja my-unzip.

1.1 My-Cat

Ohjelma lukee annettujen tiedostojen sisällön ja tulostaa sen standard-tulosteeseen. Jos tiedostojen avaaminen epäonnistuu, ohjelma tulostaa virheilmoituksen ja palaa virhekoodilla.

1.1.1 Tiedostot ja Funktiot

readfiles

Lukea kaikkien annettujen tiedostojen sisältö ja tulostaa se standard-tulosteeseen.

Parametrit:

- `char **files`: Taulukko tiedostojen poluista.
- `int num _ files`: Tiedostojen lukumäärä.

Palauttaa:

- 0 jos kaikki tiedostot luettiin onnistuneesti.
- 1 jos jonkin tiedoston avaaminen epäonnistui.

1.1.2 Pääohjelma (main)

Tarkistaa komentorivin argumentit ja kutsua `readfiles`-funktiota tiedostojen lukemiseksi.

Parametrit:

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

Toiminta:

- Jos argumentteja on alle kaksi, ohjelma lopetetaan nollalla.
- Muutoin kutsutaan `readfiles`-funktiota ja palautetaan sen paluuarvo.

```
systemiohjelmointi git:(project-2/0) gcc -o wcat my-cat.c -Wall -Werror
systemiohjelmointi git:(project-2/0) * ./wcat sample_output.txt
|
| world
| hello
* systemiohjelmointi git:(project-2/0) *
```

Kuva 1: My-cat käyttö

1.1.3 Kääntäminen ja Suorittaminen

Koodin kääntäminen ohjeistuksen mukaisesti:

```
gcc -o wcat my-cat.c -Wall -Werror
```

Ohjelman suorittaminen:

```
./wcat file1.txt file2.txt ...
```

1.1.4 Virheenkäsittely

Ohjelma käsittelee virheitä tiedostojen avaamisessa. Jos tiedoston avaaminen epäonnistuu, se tulostaa virheilmoituksen `stdout`-virtaan ja palauttaa arvon 1.

```
→ wcat git:(master) * cp ../../../../systemiohjelmointi/wcat .
→ wcat git:(master) * sh test-wcat.sh
test 1: passed
test 2: passed
test 3: passed
test 4: passed
test 5: passed
test 6: passed
test 7: passed
→ wcat git:(master) *
```

Kuva 2: My-cat testit

1.1.5 Testit

Ohjelma on läpäissyt kaikki Ostep-projektin testit (2). Ohjelmaa kutsutaan **wcat** testien läpäisemiseen.

1.2 My-grep

Ohjelma etsii tiettyä merkkijonoa annetuista tiedostoista ja tulostaa rivit, jotka sisältävät kyseisen merkkijonon. Jos tiedostoja ei ole määritetty, ohjelma lukee standard-syötteestä.

1.2.1 Tiedostot ja Funktiot

`search_in_file`

Suorittaa haku annetussa tiedostossa ja tulostaa rivit, jotka sisältävät hakuehdon.

Parametrit:

- `FILE *file`: Tiedoston osoitin.
- `const char *search_term`: Haettava merkkijono.

1.2.2 Pääohjelma (main)

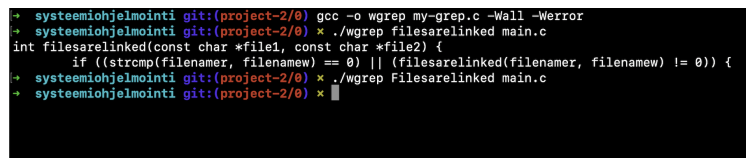
Tarkistaa komentorivin argumentit ja kutsua `search_in_file`-funktia tiedostojen hakemisen suorittamiseksi.

Parametrit:

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

Toiminta:

- Jos argumentteja on alle kaksi, ohjelma tulostaa käyttöohjeen ja lopettaa virhekoodilla 1.
- Jos vain hakuehto on määritetty, ohjelma lukee standard-syötteestä.
- Jos tiedostoja on määritetty, ohjelma käsittelee jokaisen tiedoston erikseen.



```

→ systeemiohjelmointi git:(project-2/0) gcc -o wgrep my-grep.c -Wall -Werror
→ systeemiohjelmointi git:(project-2/0) x ./wgrep filesarelinked main.c
int filesarelinked(const char *file1, const char *file2) {
    if ((strcmp(filenameer, filenameew) == 0) || (filesarelinked(filenameer, filenameew) != 0)) {
→ systeemiohjelmointi git:(project-2/0) x ./wgrep Filesarelinked main.c
→ systeemiohjelmointi git:(project-2/0) x

```

Kuva 3: My-grep käyttö

1.2.3 Kääntäminen ja Suorittaminen

Koodin kääntäminen:

```
gcc -o wgrep my-grep.c -Wall -Werror
```

Ohjelman suorittaminen:

```
./wgrep searchterm file1.txt file2.txt ..
```

1.2.4 Virheenkäsittely

Ohjelma käsittelee virheitä tiedostojen avaamisessa. Jos tiedoston avaaminen epäonnistuu, se tulostaa virheilmoituksen `stdout`-virtaan ja lopettaa tilakoodilla 1.

1.2.5 Testit

Ohjelma on läpäissyt kaikki Ostep-projektin testit (4). Ohjelmaa kutsutaan **wgrep** testien läpäisemiseen.

```
→ wget git:(master) * cp ../../../../systemiohjelmointi/wgrep .
→ wget git:(master) * sh test-wgrep.sh
test 1: passed
test 2: passed
test 3: passed
test 4: passed
test 5: passed
test 6: passed
test 7: passed
→ wget git:(master) * █
```

Kuva 4: My-grep testit

1.3 My-zip

Ohjelma suorittaa RLE (Run-Length Encoding) -koodausta annetuille tiedostoille ja tulostaa koodatun sisällön standard-tulosteeseen. Ohjelma käsittelee useita tiedostoja, jos ne annetaan komentoriviparametreina.

1.3.1 Tiedostot ja Funktiot

struct token

Rakenteen avulla tallennetaan RLE-koodauksen tiedot.

Kentät:

- `int l`: Toiston pituus.
- `char c`: Toistettava merkki.

write_rle

Kirjoittaa RLE-objekti standard-tulosteeseen.

Parametrit:

- `struct token rleobj`: RLE-objekti, joka sisältää toistopituuden ja merkin.

process

Käsitellä tiedosto ja päivittää RLE-objekti.

Parametrit:

- `FILE *stream`: Tiedostovirran osoitin.
- `struct token prev`: Edellinen RLE-objekti.

Palauttaa:

- `struct token`: Päivitetty RLE-objekti.

1.3.2 Pääohjelma (main)

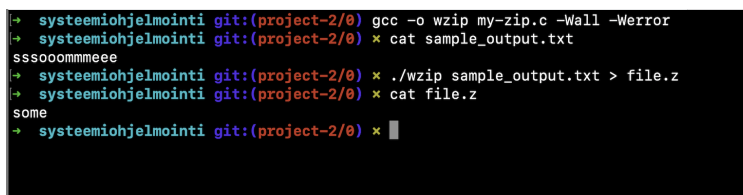
Tarkistaa komentorivin argumentit ja kutsua `process`-funktia annettujen tiedostojen koodaamiseksi.

Parametrit:

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

Toiminta:

- Jos argumentteja on alle kaksi, ohjelma tulostaa käyttöohjeen ja lopettaa virhekoodilla 1.
- Muutoin ohjelma käsittelee jokaisen tiedoston erikseen kutsuen `process`-funktia.



```
→ systeemiohjelmointi git:(project-2/0) gcc -o wzip my-zip.c -Wall -Werror
→ systeemiohjelmointi git:(project-2/0) x cat sample_output.txt
sssooommmeee
→ systeemiohjelmointi git:(project-2/0) x ./wzip sample_output.txt > file.z
→ systeemiohjelmointi git:(project-2/0) x cat file.z
some
→ systeemiohjelmointi git:(project-2/0) x
```

Kuva 5: My-zip käyttö

1.3.3 Kääntäminen ja Suorittaminen

Koodin kääntäminen:

```
gcc -o wzip my-zip.c -Wall -Werror
```

Ohjelman suorittaminen:

```
./wzip file1 file2 ...
```

1.3.4 Virheen käsittely

Ohjelma käsittelee virheitä tiedostojen avaamisessa ja RLE-koodauksen käsittelyssä. Jos tiedoston avaaminen epäonnistuu, se tulostaa virheilmoituksen `stdout`-virtaan ja lopettaa tilakoodilla 1. Ohjelma varmistaa, että viimeinen RLE-objekti kirjoitetaan standarditulosteeseen.

1.3.5 Testit

Ohjelma on läpäissyt kaikki Ostep-projektin testit (6). Ohjelmaa kutsutaan **wzip** testien läpäisemiseen.

```

→ initial-utilities git:(master) × cd wzip
→ wzip git:(master) × cp ../../../../systemiohjelmointi/wzip .
→ wzip git:(master) × sh test-wzip.sh
test 1: passed
test 2: passed
test 3: passed
test 4: passed
test 5: passed
test 6: passed
→ wzip git:(master) × █

```

Kuva 6: My-zip testit

1.4 My-unzip

Oohjelma purkaa RLE (Run-Length Encoding) -koodattua dataa ja tulostaa puretun sisällön standard-tulosteeseen. Ohjelma käsittelee useita tiedostoja, jos ne annetaan komentoriviparametreina.

1.4.1 Tiedostot ja Funktiot

decompress

Purkaa RLE-koodattua dataa annetusta tiedostovirrasta ja tulostaa sen standard-tulosteeseen.

Parametrit:

- FILE *stream: Tiedostovirran osoitin.

1.4.2 Pääohjelma (main)

Tarkistaa komentorivin argumentit ja kutsua `decompress`-funktiota annettujen tiedostojen purkamiseksi.

Parametrit:

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

Toiminta:

- Jos argumentteja on alle kaksi, ohjelma tulostaa käyttöohjeen ja lopettaa virhekoodilla 1.
- Muutoin ohjelma käsittelee jokaisen tiedoston erikseen kutsuen `decompress`-funktiota.

1.4.3 Kääntäminen ja Suorittaminen

Koodin kääntäminen:

```
gcc -o wunzip my-unzip.c -Wall -Werror
```

Ohjelman suorittaminen:

```
./wunzip file1 file2 ...
```

```

→ systeemiohjelmointi git:(project-2/0) gcc -o wunzip my-unzip.c -Wall -Werror
→ systeemiohjelmointi git:(project-2/0) * ./wunzip file.z
sssooommeeee
→ systeemiohjelmointi git:(project-2/0) * █

```

Kuva 7: My-unzip käyttö

1.4.4 Virheen käsittely

Ohjelma käsittelee virheitä tiedostojen avaamisessa ja RLE-formaatin purkamisessa. Jos tiedoston avaaminen epäonnistuu, se tulostaa virheilmoituksen `stderr`-virtaan ja lopettaa tilakoodilla 1. Jos RLE-data on virheellistä, ohjelma tulostaa virheilmoituksen ja lopettaa.

```

→ wunzip git:(master) * cp ../../../../systeemiohjelmointi/wunzip .
→ wunzip git:(master) * sh test-wunzip.sh
test 1: passed
test 2: passed
test 3: passed
test 4: passed
test 5: passed
test 6: passed
→ wunzip git:(master) * █

```

Kuva 8: My-unzip testit

1.4.5 Testit

Ohjelma on läpäissyt kaikki Ostep-projektin testit (8). Ohjelmaa kutsutaan **wunzip** testien läpäisemiseen.