



Open your mind. LUT.

Lappeenranta University of Technology

Käyttöjärjestelmät ja systeemiohjelmointi

## **Project 1 UNIX UTILITIES**

Lappeenrannan teknillinen yliopisto

Tietotekniikan koulutusohjelma

11.12.2018

CT30A3370 Käyttöjärjestelmät ja systeemiohjelmointi

Juho Kontiainen 0503209

Jesse Peltola 0523140

**Sisällysluettelo**

1. Johdanto .....	3
2. My-cat .....	3
3. My-grep .....	5
4. Yhteenveto .....	6

# 1. Johdanto

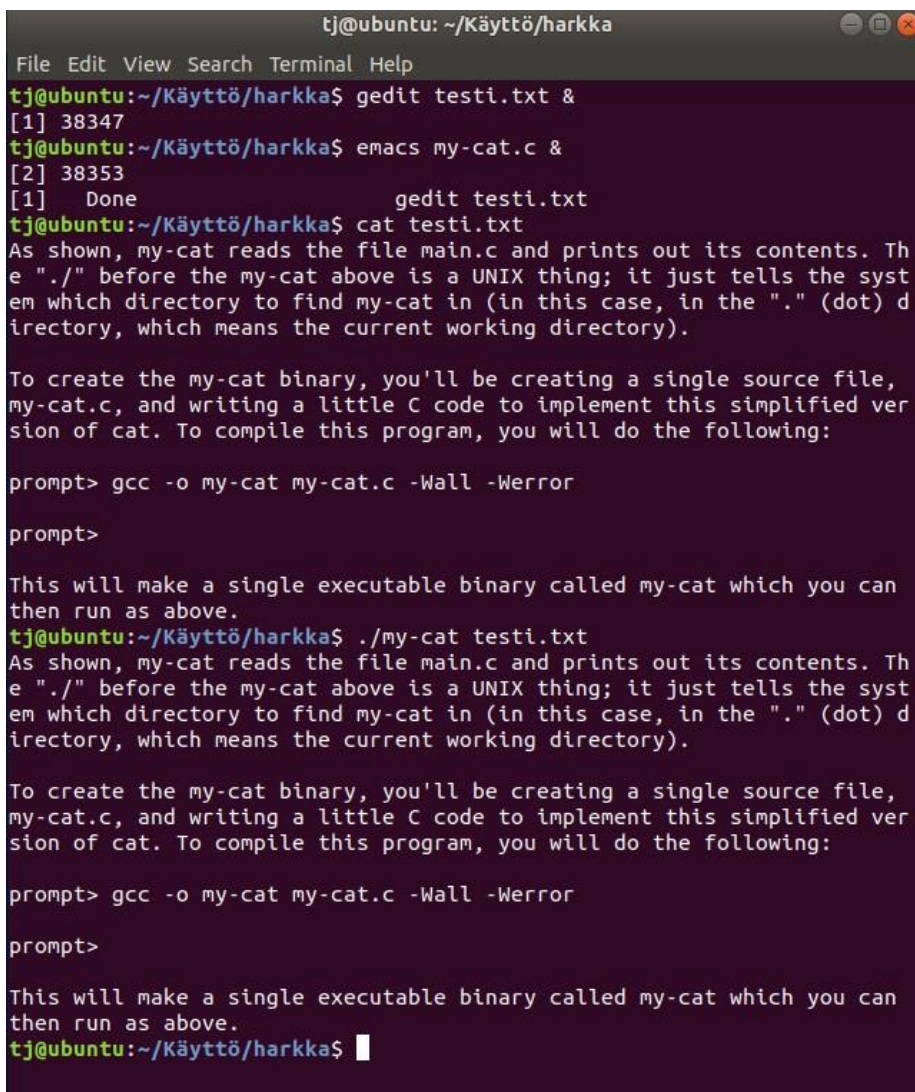
Harjoitustyössä oli tarkoituksena luoda unixin cat, grep sekä zip ja unzip-komentoja vastaavat omat versio. Me teimme unixin cat- sekä grep-komennoista omat versiot, mutta my-zip ja my-unzip jäivät toteuttamatta.

Kaikki ohjelmakoodit löytyvät githubista: <https://github.com/JuhoKon/Project1>. Kaikki palautetut koodit kääntyvät optioilla -Wall ja -Werror, sekä toimivat kuten tässä dokumentaatiossa on selostettu.

## 2. My-cat

Cat -komento tulostaa tiedoston sisällön terminaaliin.

My-cat -ohjelman toiminnallisuutta kuvattu oheisilla kuvankaappauksilla:



```
tj@ubuntu: ~/Käyttö/harkka
File Edit View Search Terminal Help
tj@ubuntu:~/Käyttö/harkka$ gedit testi.txt &
[1] 38347
tj@ubuntu:~/Käyttö/harkka$ emacs my-cat.c &
[2] 38353
[1] Done
tj@ubuntu:~/Käyttö/harkka$ cat testi.txt
As shown, my-cat reads the file main.c and prints out its contents. The
"." before the my-cat above is a UNIX thing; it just tells the system
which directory to find my-cat in (in this case, in the "." (dot) directory,
which means the current working directory).

To create the my-cat binary, you'll be creating a single source file,
my-cat.c, and writing a little C code to implement this simplified version
of cat. To compile this program, you will do the following:

prompt> gcc -o my-cat my-cat.c -Wall -Werror

prompt>

This will make a single executable binary called my-cat which you can
then run as above.
tj@ubuntu:~/Käyttö/harkka$ ./my-cat testi.txt
As shown, my-cat reads the file main.c and prints out its contents. The
"." before the my-cat above is a UNIX thing; it just tells the system
which directory to find my-cat in (in this case, in the "." (dot) directory,
which means the current working directory).

To create the my-cat binary, you'll be creating a single source file,
my-cat.c, and writing a little C code to implement this simplified version
of cat. To compile this program, you will do the following:

prompt> gcc -o my-cat my-cat.c -Wall -Werror

prompt>

This will make a single executable binary called my-cat which you can
then run as above.
tj@ubuntu:~/Käyttö/harkka$
```

Kuva 1.

Terminaalista kuvankaappaus (Kuva 1.), josta näkee käännösoptiot, oikean cat-ohjelman toiminnallisuutta sekä my-cat -ohjelman toiminnallisuutta.

```
tj@ubuntu:~/Käyttö/harkka$ cat testi.txt testi2.txt
As shown, my-cat reads the file main.c and prints out its contents. The "." before the my-cat above is a UNIX thing; it just tells the system which directory to find my-cat in (in this case, in the "." (dot) directory, which means the current working directory).

To create the my-cat binary, you'll be creating a single source file, my-cat.c, and writing a little C code to implement this simplified version of cat. To compile this program, you will do the following:

prompt> gcc -o my-cat my-cat.c -Wall -Werror

prompt>

This will make a single executable binary called my-cat which you can then run as above.
this
is
test2
tj@ubuntu:~/Käyttö/harkka$ ./my-cat testi.txt testi2.txt
As shown, my-cat reads the file main.c and prints out its contents. The "." before the my-cat above is a UNIX thing; it just tells the system which directory to find my-cat in (in this case, in the "." (dot) directory, which means the current working directory).

To create the my-cat binary, you'll be creating a single source file, my-cat.c, and writing a little C code to implement this simplified version of cat. To compile this program, you will do the following:

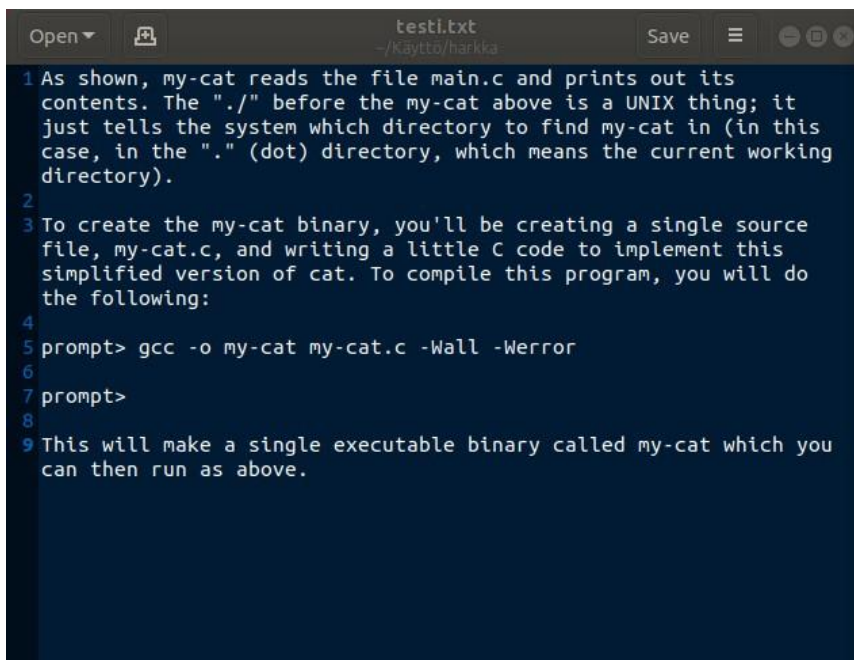
prompt> gcc -o my-cat my-cat.c -Wall -Werror

prompt>

This will make a single executable binary called my-cat which you can then run as above.
this
is
test2
tj@ubuntu:~/Käyttö/harkka$
```

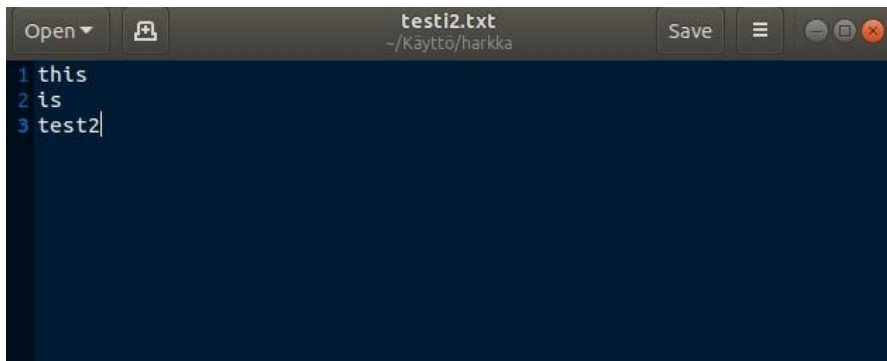
Kuva 2.

Toinen kuvankaappaus (Kuva 2.) terminaalista, josta näkee, että ohjelmaan voi syöttää monia tiedostoja.



```
1 As shown, my-cat reads the file main.c and prints out its
  contents. The "." before the my-cat above is a UNIX thing; it
  just tells the system which directory to find my-cat in (in this
  case, in the "." (dot) directory, which means the current working
  directory).
2
3 To create the my-cat binary, you'll be creating a single source
  file, my-cat.c, and writing a little C code to implement this
  simplified version of cat. To compile this program, you will do
  the following:
4
5 prompt> gcc -o my-cat my-cat.c -Wall -Werror
6
7 prompt>
8
9 This will make a single executable binary called my-cat which you
  can then run as above.
```

Kuva 3. testi.txt



Kuva 4. testi2.txt

### 3. My-grep

Grep -komennolla tulostetaan tiedostosta tietyt rivit, joista löytyy syötetty avainsana.

My-grep -ohjelman toiminnallisuutta kuvattu oheisilla kuvankaappauksilla (Kuva 5. Kuva 6.):

```
tj@ubuntu:~/Käyttö/harkka$ ./my-grep this testi.txt testi2.txt
As shown, my-cat reads the file main.c and prints out its contents. The "/" before the my-cat above is a UNIX thing; it just tells the system which directory to find my-cat in (in this case, in the "." (dot) directory, which means the current working directory).
To create the my-cat binary, you'll be creating a single source file, my-cat.c, and writing a little C code to implement this simplified version of cat. To compile this program, you will do the following:
this
tj@ubuntu:~/Käyttö/harkka$ grep this testi.txt testi2.txt
testi.txt:As shown, my-cat reads the file main.c and prints out its contents. The "/" before the my-cat above is a UNIX thing; it just tells the system which directory to find my-cat in (in this case, in the "." (dot) directory, which means the current working directory).
testi.txt:To create the my-cat binary, you'll be creating a single source file, my-cat.c, and writing a little C code to implement this simplified version of cat. To compile this program, you will do the following:
testi2.txt:this
tj@ubuntu:~/Käyttö/harkka$
```

Kuva 5.

```
tj@ubuntu:~/Käyttö/harkka$ grep "" testi.txt
As shown, my-cat reads the file main.c and prints out its contents. The "/" before the my-cat above is a UNIX thing; it just tells the system which directory to find my-cat in (in this case, in the "." (dot) directory, which means the current working directory).

To create the my-cat binary, you'll be creating a single source file, my-cat.c, and writing a little C code to implement this simplified version of cat. To compile this program, you will do the following:

prompt> gcc -o my-cat my-cat.c -Wall -Werror

prompt>

This will make a single executable binary called my-cat which you can then run as above.
tj@ubuntu:~/Käyttö/harkka$
```

Kuva 6.

My-grep tulostaa etsityn avainsanan sisältävät rivit molemmista tiedostoista kuten kuvasta 5 näkyy (Kuva 5). Vertailun vuoksi sama komento on ajettu myös unixin omalla grep -komennolla. Jos ohjelmalle syöttää tyhjän merkkijono (""), niin ohjelma tulostaa koko tiedoston sisällön. Ohjelman ajoon käytetyt testi.txt (Kuva 3) ja testi2.txt (Kuva 4)-tiedostot ovat samat kuin cat-esimerkissä käytetyt.

## 4. Yhteenveto

my-cat

- Tulostaa tiedoston sisällön terminaaliin
- Useamman tiedoston syöttäminen toimii my-grep
- Lukee tiedoston rivi riviltä ja jos löytää avainsanaa vastaavan merkkijonon, niin tulostaa tämän rivin käyttäjälle
- Toimii kuten grep -komento, mutta ei värjää haettua merkkijonoa.
- Jos rivin pituus on ongelma, niin sitä voidaan kasvattaa muuttamalla MAXBUFFER-makroa.
- Tyhjä merkkijono merkitään "", ja se annettuna grepillä tulostaa kaiken tiedoston sisällön.