

LUT School of Engineering Science

CT30A3370 Käyttöjärjestelmät ja systeemiohjelmointi

Harjoitustyö: Project 1: Unix Utilities

Project 1

Unix Utilities

16.12.2019

Tero Nevalainen 0521595

SISÄLLYSLUETTELO

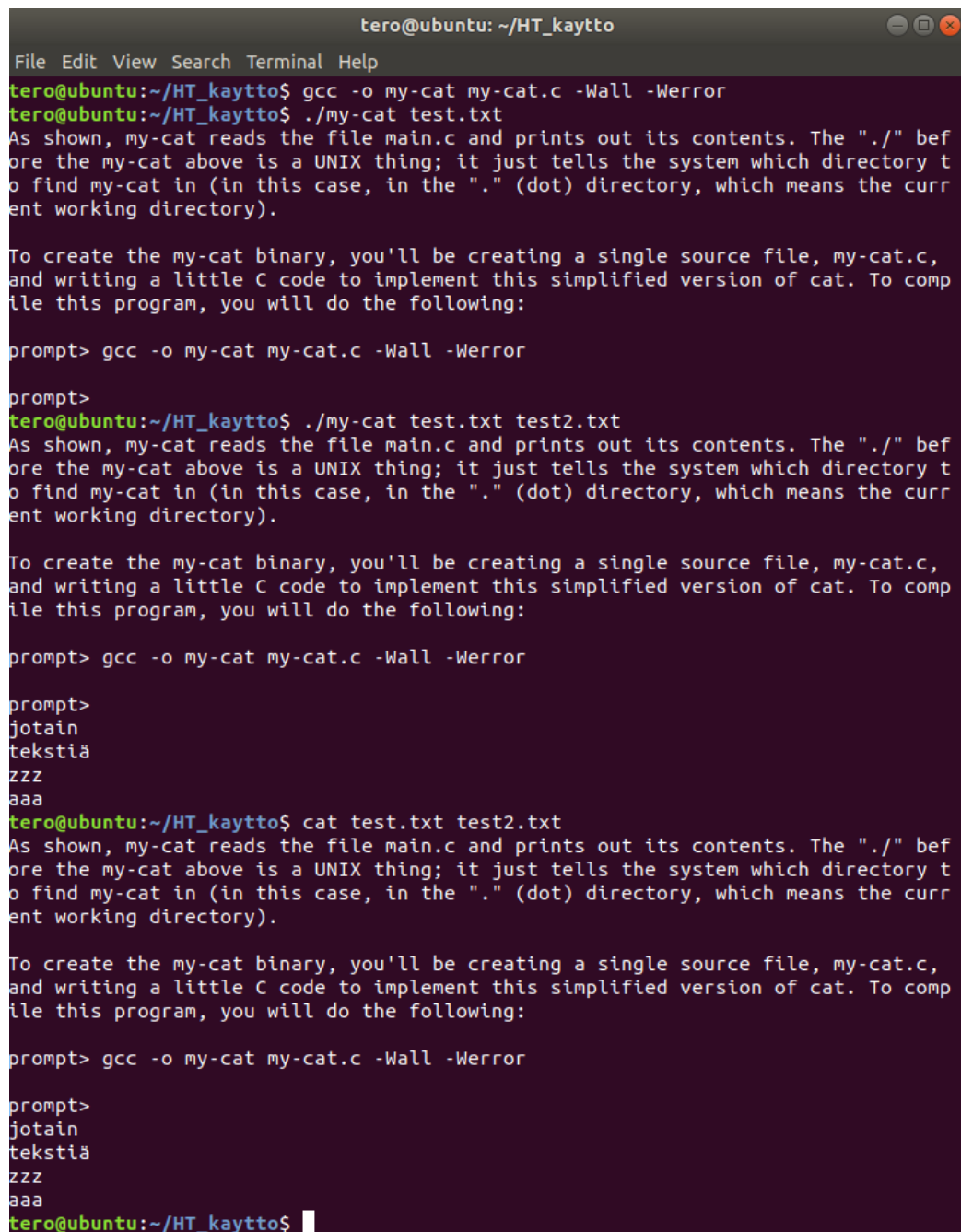
1	Johdanto	1
2	Toiminta.....	Virhe. Kirjanmerkkiä ei ole määritetty.
3	Kuva toiminnasta	Virhe. Kirjanmerkkiä ei ole määritetty.

1 JOHDANTO

Tehtävänä oli luoda my-cat ja my-grep ohjelmat, jotka löytyvät <https://github.com/teronevalainen/Project1-Unix-utilities>. My-zip ja my-unzip puuttuvat.

2 MY-CAT

Cat tulostaa terminaalin tiedoston sisällön. My-cat ohjelman toiminnasta kuva:



```
tero@ubuntu: ~/HT_kaytto
File Edit View Search Terminal Help
tero@ubuntu:~/HT_kaytto$ gcc -o my-cat my-cat.c -Wall -Werror
tero@ubuntu:~/HT_kaytto$ ./my-cat test.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>
tero@ubuntu:~/HT_kaytto$ ./my-cat test.txt test2.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>
jotain
tekstiä
zzz
aaa
tero@ubuntu:~/HT_kaytto$ cat test.txt test2.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).

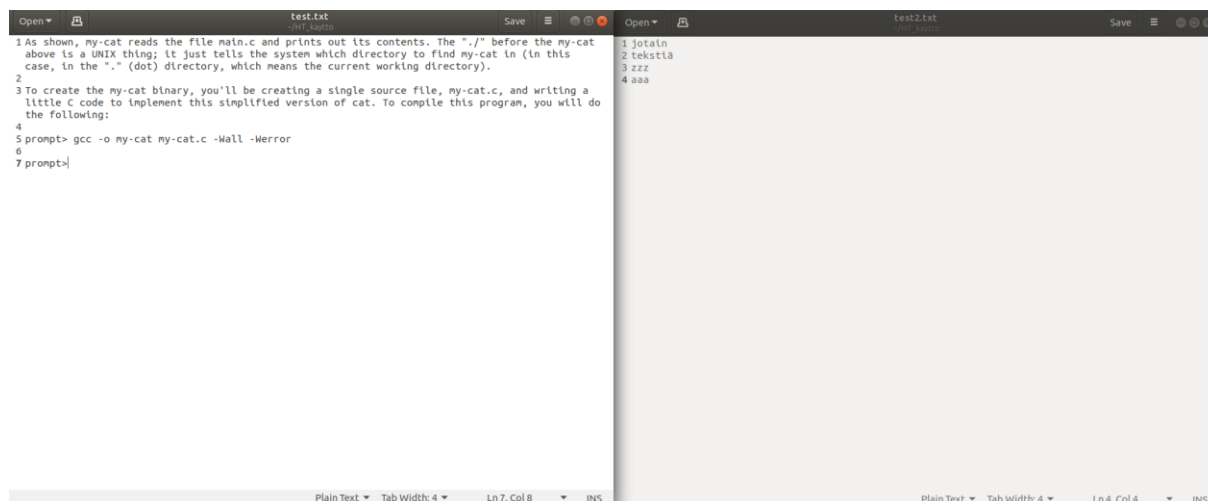
To create the my-cat binary, you'll be creating a single source file, my-cat.c, and writing a little C code to implement this simplified version of cat. To compile this program, you will do the following:

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

prompt>
jotain
tekstiä
zzz
aaa
tero@ubuntu:~/HT_kaytto$
```

Kuva 1

My-cat kääntyy komennolla `gcc -o my-cat my-cat.c -Wall -Werror`. (Kuva 1) ensiksi tulostetaan `test.txt` tiedoston sisältö, jonka jälkeen tulostetaan molempien `test.txt` ja `test2.txt` tiedostojen sisältö. Toiminnan varmistamiseksi tulostin myös tiedostot komennolla `cat test.txt test2.txt`, mistä huomaa täsmälleen saman tulostuksen. Alla kuvat tekstiedostoista (kuva 2) `test.txt` ja `test2.txt`.



Kuva2

```
tero@ubuntu:~/HT_kaytto$ ./my-cat my-cat.c
/* Tero Nevalainen */
/*My-cat.c*/
/*Lähteet: https://www.tutorialspoint.com/c_standard_library/c_function_fgets.htm
man sivut*/

#include <stdio.h>
#include <stdlib.h>
int cat(char *);

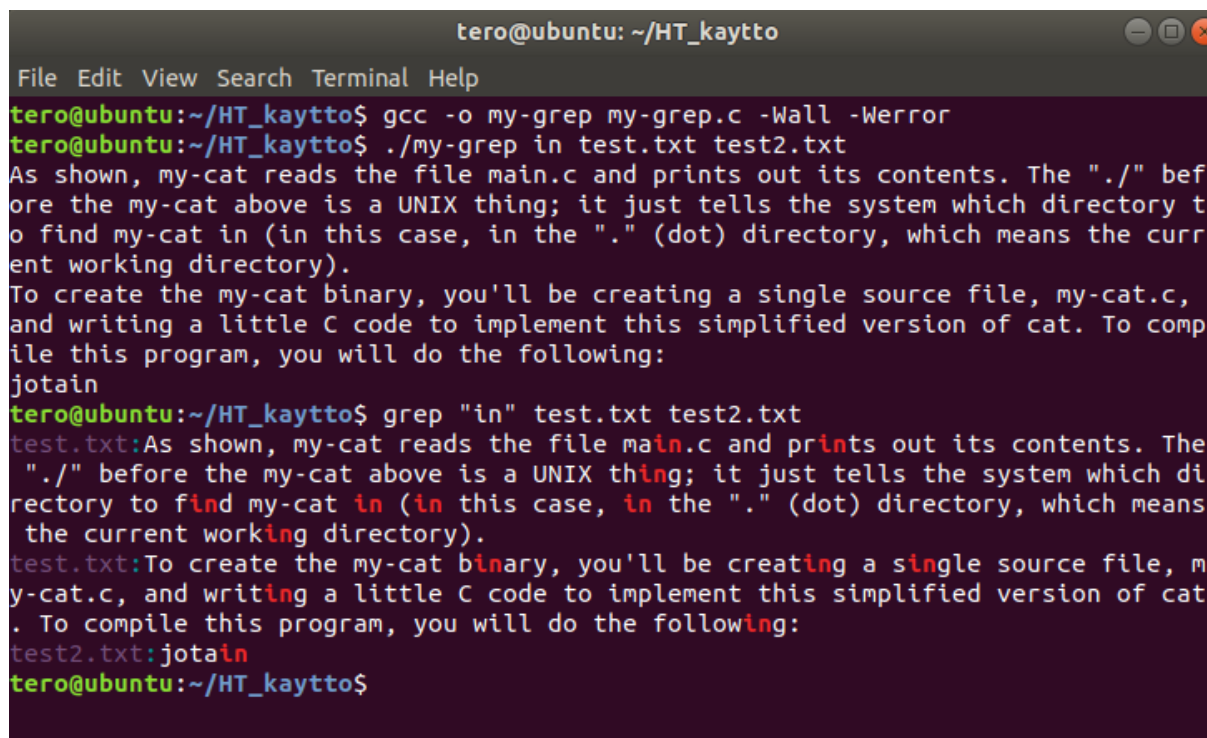
int main(int argc, char *argv[]) {
    int i;
    if (argc <= 1) {
        return(1);
    }
    for (i=1; i<argc; i++) {
        cat(argv[i]);
    }
    return(0);
}

int cat(char *argv) {
    char x[100];
    int koko = 100;
    FILE *fp;
    if ((fp = fopen(argv, "r")) == NULL) {
        printf("My-cat: Cannot open file\n");
        exit(1);
    }
    while (fgets(x, koko, fp)) {
        printf("%s", x);
    }
    fclose(fp);
    return(0);
}
tero@ubuntu:~/HT_kaytto$
```

Kuva 3

3 MY-GREP

Grep etsii tiedostoista säännöllisillä lausekkeilla määriteltyjä merkkijonoja. My-grep käänytyy komennolla `gcc -o my-grep my-grep.c -Wall -Werror`. Ohessa grepin toimintaa:



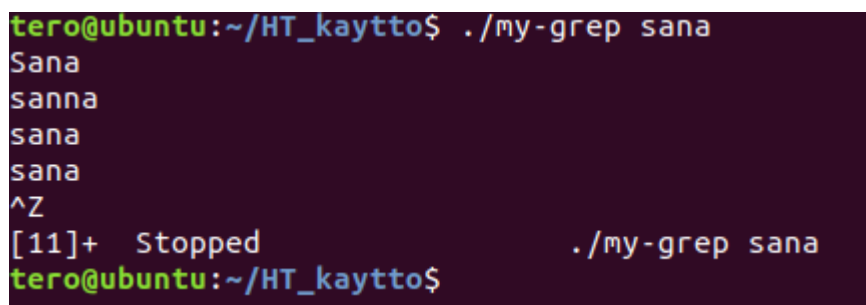
```

tero@ubuntu: ~/HT_kaytto
File Edit View Search Terminal Help
tero@ubuntu:~/HT_kaytto$ gcc -o my-grep my-grep.c -Wall -Werror
tero@ubuntu:~/HT_kaytto$ ./my-grep in test.txt test2.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:
jotain
tero@ubuntu:~/HT_kaytto$ grep "in" test.txt test2.txt
test.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).
test.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:
test2.txt:jotain
tero@ubuntu:~/HT_kaytto$

```

Kuva 4

Tekstitiedostot ovat samoja kuin my-catissa. Kuvasta huomaa, kuinka grep etsii "in" merkkijonoja ja tulostaa lauseet, joissa se esiintyy. My-grep ei värjää sanoja niin kuin grep tekee.



```

tero@ubuntu:~/HT_kaytto$ ./my-grep sana
Sana
sanna
sana
sana
^Z
[11]+  Stopped                  ./my-grep sana
tero@ubuntu:~/HT_kaytto$

```

Kuva 5

Tässä my-grep etsii standard inputista "sana" ja printtaa "sana" jos sen kirjoittaa ohjelmaan, siksi sana on kaksi kertaa peräkkäin ohjelmassa syötteiden "Sana" ja "sanna" jälkeen.