

# Käyttöjärjestelmät ja systemiohjelmointi

Kurssin harjoitustyöraportti

# Sisällysluettelo

<b>Sisällysluettelo</b>	<b>1</b>
<b>Projekti 1</b>	<b>2</b>
<b>Projekti 2</b>	<b>4</b>
<b>Projekti 4</b>	<b>6</b>
<b>Tehtävien GitHub Repositorio</b>	<b>7</b>

# Projekti 1

<https://github.com/TepaXD/HarkkaRepo/tree/master/Projekti1>

Tehtävässä toteutettiin koodi joka muodostaa annetusta syötteestä tiedoston, jossa syötteen rivit ovat vastakkaisessa järjestyksessä.

Esim.

Input.txt ->                      Output.txt ->

Haloo	12485
Lahoo	Kaloja
Kaloja	Lahoo
12485	Haloo

Ohjelma toimii kolmella eri tapaa annetuista komentorivi argumenteista riippuen.

Nolla argumenttia:

- Syöte annetaan terminaalissa, ja se tulostuu väärinpäin terminaaliin.

```
Teemu-MacBook-Pro-5:projekti1 TeemuK$ ./reverse
give input to reverse: tämä teksti peilaantuu

peilaantuu teksti tämä
Teemu-MacBook-Pro-5:projekti1 TeemuK$
```

Yhdellä argumentilla:

- Annetusta tiedostosta (input.txt esimerkissä) luetaan syöte ja tulostetaan ruudulle.

```
5555555
4444444
3333333
2222222
1111111

Teemu-MacBook-Pro-5:projekti1 TeemuK$ ./reverse input.txt

5555555
4444444
3333333
2222222
1111111
Teemu-MacBook-Pro-5:projekti1 TeemuK$
```

Kahdella argumentilla:

- Ohjelma lukee syötteen ensimmäisestä tiedostosta ja tallentaa toiseen.

```
Teemu-MacBook-Pro-5:projekti1 TeemuK$ ./reverse input2.txt output2.txt
Teemu-MacBook-Pro-5:projekti1 TeemuK$
```

```
aaaaaa
bbbbbbbbbbbbbb
cc
dddddddddddddddddd
eeeee
fff
```

```
fff
eeeeee
dddddddddddddddddd
cc
bbbbbbbbbbbbbb
aaaaaa
```

# Projekti 2

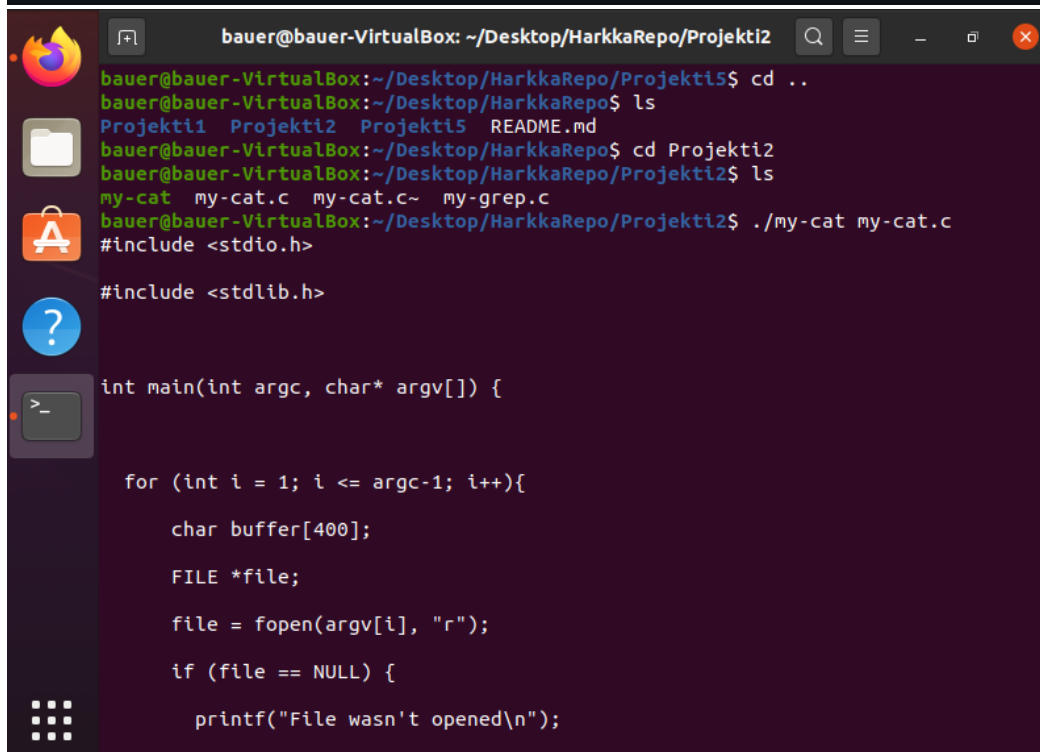
Repo: <https://github.com/TepaXD/HarkkaRepo/tree/master/Projekti2>

Tehtävä koostuu neljästä pienemmästä tehtävästä, jotka kaikki löytyy saman kansion alta.

## my-cat:

Ensimmäinen 2 projektin pienohjelmista, my-cat, lukee C-kielisen tiedoston ja tulostaa terminaaliin sisällön.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char* argv[]) {
5
6     for (int i = 1; i <= argc-1; i++){
7         char buffer[400]; //Reserving memory for the contents
8         FILE *file;
9         file = fopen(argv[i], "r"); //Getting the filename from the commandline argument and opening file
10        if (file == NULL) { //Error handling incase the file doesn't work
11            printf("File wasn't opened\n");
12            exit(1);
13        }
14
15        while(fgets(buffer, 399, file) != NULL) { //Print the contents of "buffer"
16            printf("%s\n", buffer);
17        }
18        fclose(file); //close file
19    }
20    return 0;
21 }
```



```
bauer@bauer-VirtualBox: ~/Desktop/HarkkaRepo/Projekti2
bauer@bauer-VirtualBox:~/Desktop/HarkkaRepo$ cd ..
bauer@bauer-VirtualBox:~/Desktop/HarkkaRepo$ ls
Projekti1 Projekti2 Projekti5 README.md
bauer@bauer-VirtualBox:~/Desktop/HarkkaRepo$ cd Projekti2
bauer@bauer-VirtualBox:~/Desktop/HarkkaRepo/Projekti2$ ls
my-cat my-cat.c my-cat.c~ my-grep.c
bauer@bauer-VirtualBox:~/Desktop/HarkkaRepo/Projekti2$ ./my-cat my-cat.c
#include <stdio.h>

#include <stdlib.h>

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

    for (int i = 1; i <= argc-1; i++){

        char buffer[400];

        FILE *file;

        file = fopen(argv[i], "r");

        if (file == NULL) {

            printf("File wasn't opened\n");
```

## my-grep:

Toinen pienohjelma, my-grep, hakee tiedostoista tai käyttäjän syötteestä annetun hakutermin mukaisia sanoja.

Input.txt tiedosto:

```
testi
tämä on oikea testi
tätä ei tulosteta
mutta tämä testi tulostuu!
```

Hakusanana kirjain "t":

```
[Teemu-MacBook-Pro-5:grep TeemuK$ ./my-grep t input.txt
testi

tämä on oikea testi

tätä ei tulosteta

mutta tämä testi tulostuu!
Teemu-MacBook-Pro-5:grep TeemuK$
```

Hakusanana sana "testi":

```
[Teemu-MacBook-Pro-5:grep TeemuK$ ./my-grep testi input.txt
testi

tämä on oikea testi

mutta tämä testi tulostuu!
Teemu-MacBook-Pro-5:grep TeemuK$
```

## my-zip:

## my-unzip:

## Projekti 4

Tähän ei ole vielä tehty mitään :).

# Tehtävien GitHub Repositorio

<https://github.com/TepaXD/HarkkaRepo>