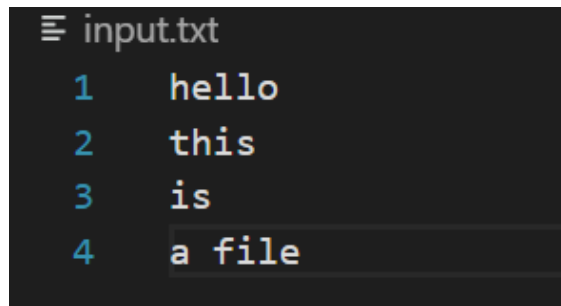


### Project 1: Warmup to C and Unix programming

Tehtävänä oli toteuttaa ohjelma *reverse*, joka lukee rivejä käyttäjän syötteestä tai tiedostosta ja tulostaa ne käänteisessä järjestyksessä. Toteutin ohjelman rakenteen hyödyntämällä rekursiivisuutta: rivien lukemisesta ja tulostamisesta vastaava aliohjelma kutsuu itseään uudelleen niin kauan kuin käyttäjän syötteessä tai tiedostossa on luettavia rivejä, jonka jälkeen aliohjelmat tulostavat lukemansa rivit ennen paluuta kutsuneeseen aliohjelmaan.

Ohjelmaa kutsuttaessa käyttäjä voi antaa nollasta kahteen argumenttia, joiden perusteella ohjelma käsittelee saamia syöterivejä. Kun ohjelmaa kutsutaan ilman argumentteja, ohjelma lukee käyttäjän syötteitä ja tulostaa nämä aiemmin määritetyllä tavalla. Kun ohjelmaa kutsutaan yhdellä argumentilla, ohjelma tulkitsee tämän argumentin luettavaksi tiedostoksi, josta syöte luetaan ja lopulta tulostetaan. Kahden argumentin tapauksessa ensimmäinen argumentti tulkitaan luettavaksi tiedostoksi ja toinen tiedostoksi, johon kirjoitetaan. Ohjelman toiminnallisuutta havainnollistetaan seuraavissa kuvissa.



```
≡ input.txt
1  hello
2  this
3  is
4  a file
```

Kuva 1: Demonstraatiossa käytetyn tiedoston sisältö, huomaa rivinvaihtomerkin puuttuminen lopusta



```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./reverse
Please enter a line (press ENTER to stop): hello
Please enter a line (press ENTER to stop): this
Please enter a line (press ENTER to stop): is
Please enter a line (press ENTER to stop): user
Please enter a line (press ENTER to stop): input !
Please enter a line (press ENTER to stop):
input !
user
is
this
hello
tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 2: Ohjelman reverse toiminta käyttäjän syötteellä

```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./reverse input.txt
a fileis
this
hello
tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 3: Ohjelman reverse toiminta tiedostoa luettaessa

```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./reverse input.txt output.txt
tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 4: Ohjelman reverse toiminta kahdella tiedostolla

```
output.txt
1  a fileis
2  this
3  hello
4
```

Kuva 5: Tiedoston output.txt sisältö

```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./reverse input.txt input.txt
Input and output file must differ.
tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 6: Ohjelman reverse toiminta virheellisessä syötteessä (tiedostojen nimet samat)

```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./reverse input.txt output.txt plusyks.txt
usage: ./reverse <input> <output>
tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 7: Ohjelman reverse virheellisessä syötteessä (liikaa argumentteja)

```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./reverse input1.txt
error: cannot open file 'input1.txt'.
tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 8: Ohjelman reverse toiminta, kun luettavaa tiedostoa ei ole olemassa

## Project 2: Unix Utilities

- **my-cat:** Tehtävänä oli kirjoittaa ohjelma, joka lukee ja tulostaa lukemiensa tiedostojen sisällön. Toteutin ohjelman tiedostonkäsittelyn C-ohjelmoinnin perusteet -kurssilla opetellulla tavalla sekä hyödyntämällä getline-funktiota, koska sen avulla pystytään lukemaan mielivaltaisen pituisia rivejä.

```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./my-cat input.txt output.txt
hello
this
is
a filea fileis
this
hello
tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 9: Ohjelman my-cat toiminta kahdella tiedostolla

- **my-grep:** Tehtävänä oli kirjoittaa ohjelma, joka etsii lukemistaan tiedostoista käyttäjän määrittämää termiä ja tulostaa nämä rivit, joista kyseinen termi löytyy. Toteutin ohjelman hyödyntämällä strstr()-funktia.

```
● tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./my-grep line reverse.c
char *line = NULL;
fprintf(stdout, "Please enter a line (press ENTER to stop): ");
if ((nread=getline(&line,&len,stdin))!=1){
    fprintf(stdout,"%s",line);
    free(line);
    char *line = NULL;
    if ((nread=getline(&line,&len,file))!=-1){
        fprintf(stdout,"%s",line);
        free(line);
        char *line = NULL;
        if ((nread=getline(&line,&len,file1))!=-1){
            fprintf(file2,"%s",line);
            free(line);
        }
    }
}
○ tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 10: Ohjelman my-grep toiminta etsiessä sanaa "line" reverse.c -tiedostosta

```
● tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./my-grep "" reverse.c
○ tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 11: Ohjelman my-grep toiminta, kun syötteessä ei ole hakutermiä

- **my-zip & my-unzip:** Tehtävänä oli kirjoittaa ohjelmat, joilla käyttäjä pystyy pakkaamaan ja purkamaan tiedostoja. Toteutin pakkaamisen lukemalla pakattavasta tiedostosta merkkejä yksi kerrallaan, laskemalla samalla peräkkäin toistuvien merkkien lukumäärää. Pakattu tiedosto kirjoitetaan binaariksi RLE-muodossa eli merkkien määrä ja itse merkki. Pakattujen tiedostojen purkaminen tapahtuu kirjoittamalla silmukassa merkkejä tiedoston osoittaman merkkien määrän perusteella.

```
● tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./my-zip input.txt > file.z
● tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./my-unzip file.z
hello
this
is
○ a filetvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 12: Tiedoston pakkaus ja purkaminen my-zip ja my-unzip ohjelmilla

### Project 3: Unix Shell

Tehtävänä oli implementoida oma komentorivitulkki *wish*, joka toteuttaa silmukassa käyttäjän syötteestä tai tiedostosta lukemiaan komentoja. Tulkkiin piti rakentaa kolme sisäänrakennettua komentoa: exit, cd ja path, loput komennoista toteutetaan execv()-funktia hyödyntäen. Toteutin ohjelman rakenteen siten, että parsin komennot hyödyntäen strtok()-funktia. Komennot tallennetaan linkitettyyn listaan, jotta niiden tallentaminen ja läpikäynti tapahtuu mahdollisimman sujuvasti. Lisäksi komentoja varten luotuun tietueeseen on määritelty jäsenmuuttujia, kuten osoittimet argumentteja varten, tieto argumenttien määrästä ja osoitin tiedostoa varten (käytetään uudelleenohjauksessa). Myös poluille on määritelty linkitetty lista ja tätä käytetään path-komennossa polkujen asettamiseen sekä execv()-funktia hyödyntävien komentojen

suorittamisessa. Komentorivitulkki voi suorittaa rinnakkaisia komentoja, sillä `execv()`-funktion komennot tapahtuvat erillisissä lapsiprosesseissa `fork()`-funktia hyödyntäen. Ohjelman toiminnallisuutta havainnollistetaan seuraavissa kuvissa.

```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./wish
wish> cd ..
Current working directory is: /home/tvk
wish> path /bin/usr /bin
Paths:
/bin/usr
/bin
wish> exit
tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 13: Ohjelman wish toiminta sisäänrakennetuilla komennoilla

```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./wish
wish> cd ..
Current working directory is: /home/tvk
wish> ls > output.txt
wish> exit
```

Kuva 14: Ohjelman wish toiminta uudelleenohjauksessa

```
output.txt
1 Ckurssi
2 ComputerScience
3 Kysteemit
4 Linux
5 Webapp
6 output.txt
7 tps.txt
8
```

Kuva 15: Tiedoston output.txt sisältö

```
tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./wish
wish> cd ..
Current working directory is: /home/tvk
wish> cat output.txt & ls & touch tmp.txt
Ckurssi ComputerScience Kysteemit Linux Webapp output.txt tmp.txt tps.txt
Ckurssi
ComputerScience
Kysteemit
Linux
Webapp
output.txt
tps.txt
wish> ls
Ckurssi ComputerScience Kysteemit Linux Webapp output.txt tmp.txt tps.txt
wish> exit
tvk@LAPTOP-PORFEVHB:~/Kysteemit$
```

Kuva 16: Ohjelman wish toiminta; useita rinnakkaisia komentoja

```

● tvk@LAPTOP-PORFEVHB:~/Kysteemit$ ./wish commands.txt
Current working directory is: /home/tvk
Ckurssi ComputerScience Kysteemit Linux Webapp output.txt tmp.txt tps.txt
Paths:
/usr/bin
Current working directory is: /home/tvk/Ckurssi
HT          HTPerustaso.c IO.h      L01-T4.c   L01T4      L02-T4.c   L03-T1.c   L03T1
HT.o        HTPerustaso.o IO.o      L01-T5.c   L01T5      L02-T5.c   L03-T2.c   L03T2
HTPerusKirjasto.c HTajo      L01-T1.c   L01T1      L02-T1.c   L02T2      L03-T3.c   L03T3
HTPerusKirjasto.h HTtesti    L01-T2.c   L01T2      L02-T2.c   L02T3      L03-T4.c   L03T4
HTPerusKirjasto.o IO.c       L01-T3.c   L01T3      L02-T3.c   L02T4      L03-T5.c   L03T5
#include <stdio.h>
int main(void) {
    printf("Ensimmäinen oma C-ohjelma. \n");
    return(0);
}
○ tvk@LAPTOP-PORFEVHB:~/Kysteemit$ 

```

Kuva 17: Ohjelman wish toiminta; komentojen suorittaminen tiedostosta

```

≡ commands.txt
1  cd ..
2  ls
3  path /usr/bin
4  cd Ckurssi
5  ls & cat L01-T1.c
6  exit

```

Kuva 18: Tiedoston commands.txt sisältö

Lähteet:

Nikula, Uolevi (2021). C-ohjelmointiopas versio 2.1 LUT Scientific and Expertise Publications Oppimateriaalit - Lecture Notes 24. LUT University.