

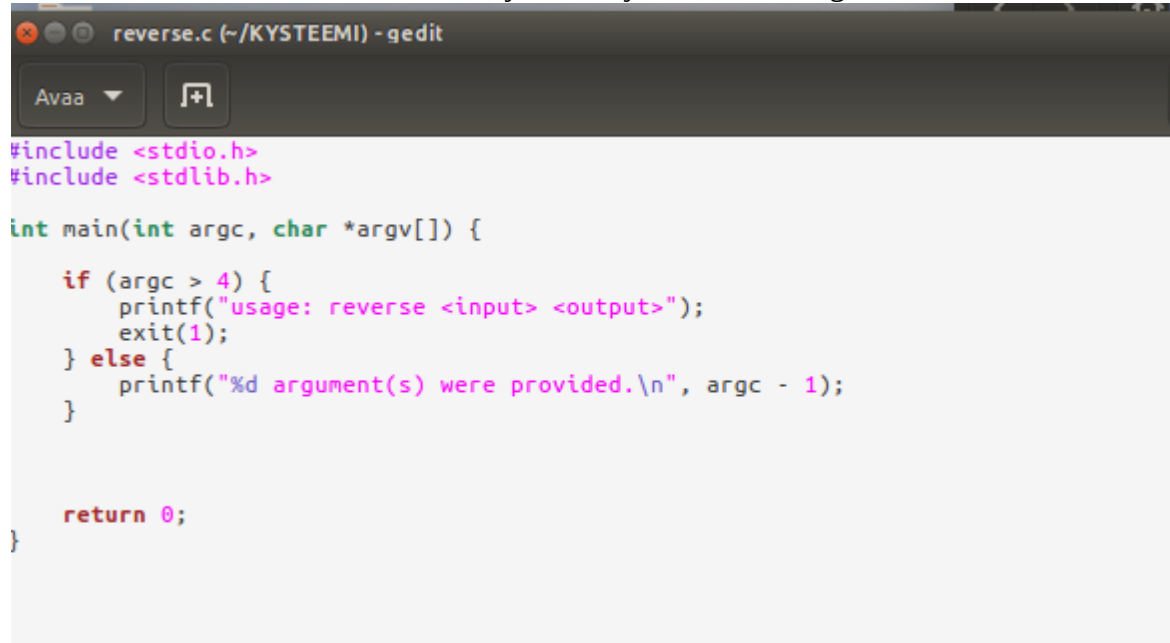
Käyttöjärjestelmät ja systeemiohjelmointi kesä 2024

Harjoitustyöprojekti

Pahoittelen tiedoston huonolaatuisia kuvakaappauksia. Näytöillä näkyvät laatikot eivät ole vika vaan ominaisuus vanhassa Ubuntussani. Sain loppujen lopuksi ongelman korjattua.

Project 1: Warmup to C and Unix programming

Aloitetaan luomalla toimiva alku C-ohjelmalle, joka tarkistaa argumenttien määrän:



```
reverse.c (~/KYTEEMI) - gedit
Avaa ▾ [icon]

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc > 4) {
        printf("usage: reverse <input> <output>");
        exit(1);
    } else {
        printf("%d argument(s) were provided.\n", argc - 1);
    }

    return 0;
}
```

Sitten teemme ohjelman, joka input-tiedoston saadessaan onnistuu lukemaan sen stdoutiin:

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc > 4) {
        printf("usage: reverse <input> <output>");
        exit(1);
    } else if (argc == 2) {
        // This is if only inputfile is given

        FILE *inputFile = fopen(argv[1], "r");
        if (inputFile == NULL) {
            perror("Error opening input file");
            exit(1);
        }

        char line[256];
        while (fgets(line, sizeof(line), inputFile)) {
            printf("%s", line);
        }

        fclose(inputFile);
    } else {
        printf("%d argument(s) were provided.\n", argc - 1);
    }

    return 0;
}

```

Joka näyttää toimivan:

```

markus@markus-X550CL:~/KYSTEMI$ ./rev input.txt
hello
i
am
an
engineer

```

(olen kirjautuneena isäni tunnuksilla, enkä jaksa siirtyä enää omilleni, joten teen täällä, ihan itse olen tämän kaiken kuitenkin tehnyt)

Nyt muokataan luku käyttämään linkitettyä listaa niin, että tiedosto voi olla kuinka pitkä tahansa. Lisätään myös jokainen alkio aina listan alkuun niin se saadaan tulostumaan käänteisenä.

Linkitettyä listaa varten luodaan seuraavat:

```

typedef struct Node {
    char line[256];
    struct Node *next;
} Node;

// Makes a new node
Node* createNode(char *line) {
    Node *newNode = (Node*)malloc(sizeof(Node));
    strcpy(newNode->line, line);
    newNode->next = NULL;
    return newNode;
}

// Adds the new node to the beginning of the linked list
void addNodeToBeginning(Node **head, char *line) {
    Node *newNode = createNode(line);
    newNode->next = *head;
    *head = newNode;
}

// Frees memory
void freeMemory(Node *head) {
    Node *temp;
    while (head != NULL) {
        temp = head;
        head = head->next;
        free(temp);
    }
}

```

Asetettiin riville kumminkin staattinen maksimimitta, koska linkitetyn listan tekeminen linkitetyn listan sisälle ei tunnu järkevältä.

```

markus@markus-X550CL:~/KYTEEMI$ gcc reverse.c -o rev
markus@markus-X550CL:~/KYTEEMI$ ./rev input.txt
engineer
an
an
i
hello

```

Näyttää toimivan.

Seuraavaksi tehtävänannosta löytyi hyvä vinkki käyttää getline() komentoa, jotta voidaan lukea minkä tahansa mittainen rivi.

```

// For dynamic reading of the inputfile using getline s.t. the line can be any length
Node *head = NULL;
char *line = NULL;
size_t len = 0;
ssize_t read;

while ((read = getline(&line, &len, inputFile)) != -1) {
    addNodeToBeginning(&head, line);
}

free(line);
fclose(inputFile);

```

Node-structia piti vähän muuttaa, rivistä tehdään pointeri:

```

typedef struct Node {
    char *line;
    struct Node *next;
} Node;

```

Tämän jälkeen koodin rakennetta muutettiin niin, että inputFile ja outputFile alustetaan NULL:ksi ja annettujen parametrien mukaisesti määritetään joko annetuksi tiedostoksi tai stdin/stdout. Tämän lisäksi käydään läpi tehtävänannon tarkempia vaatimuksia errorista yms formaliteeteista.

```

int main(int argc, char *argv[]) {
    // If too many arguments
    if (argc > 3) {
        fprintf(stderr, "usage: reverse <input> <output>\n");
        exit(1);
    } else if (argc == 3) { // If the input and output files are the same
        if (strcmp(argv[1], argv[2]) == 0) {
            fprintf(stderr, "Input and output file must differ\n");
        }
    }

    // Initialize files to be NULL
    FILE *inputFile = NULL;
    FILE *outputFile = NULL;

    if (argc > 1) { // If arguments are given, the first one is the input file
        inputFile = fopen(argv[1], "r");
        if (inputFile == NULL) {
            fprintf(stderr, "error: cannot open file '%s'\n", argv[1]);
            exit(1);
        }
    } else { // If no arguments, reads from stdin
        inputFile = stdin;
    }

    if (argc == 3) { // If 3 arguments, the third one is the output file
        outputFile = fopen(argv[2], "w");
        if (outputFile == NULL) {
            fprintf(stderr, "error: cannot open file '%s'\n", argv[2]);
            fclose(inputFile);
            exit(1);
        }
    } else { // Otherwise print to stdout
        outputFile = stdout;
    }

    // Outputting
    Node *current = head;
    while (current != NULL) {
        fprintf(outputFile, "%s", current->line);
        current = current->next;
    }
}

```

Sitten testataan, toimiiko ilman annettuja parametreja:

```

markus@markus-X550CL:~/KYTEEMI$ ./rev
oh
yeah
i
think
this
actually
works
works
actually
this
think
i
yeah
oh
markus@markus-X550CL:~/KYTEEMI$

```

Sitten testataan, toimiiko pelkällä input-tiedostolla:

```

markus@markus-X550CL:~/KYTEEMI$ ./rev input.txt
engineer
an
an
i
hello
markus@markus-X550CL:~/KYTEEMI$

```

Sitten testataan kahdella eri tiedostolla:

```
markus@markus-X550CL:~/KYTEEMI$ touch output.txt
markus@markus-X550CL:~/KYTEEMI$ ./rev input.txt output.txt
markus@markus-X550CL:~/KYTEEMI$ cat output.txt
engineer
an
am
i
hello
markus@markus-X550CL:~/KYTEEMI$
```

Kaikki näyttävät toimivan. Tarkistetaan vielä, että koodi saa kiinni virheet väärästä määrästä parametreja ja siitä, jos input-ja output-tiedostot ovat samat tai jos tiedoston avaaminen ei onnistu.

```
markus@markus-X550CL:~/KYTEEMI$ ./rev input.txt output.txt input.txt
usage: reverse <input> <output>
markus@markus-X550CL:~/KYTEEMI$ ./rev input.txt input.txt
Input and output file must differ
markus@markus-X550CL:~/KYTEEMI$ ./rev inp.txt
error: cannot open file 'inp.txt'
markus@markus-X550CL:~/KYTEEMI$
```

Näyttää toimivan juuri niin kuin pitääkin.

Project 2

my-cat.c

Aloitetaan luomalla ohjelma, joka vain lukee yhden annetun tiedoston tehtävänannossa määritellyllä tavalla:

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    FILE *fp = fopen(argv[1], "r");
    if (fp == NULL) {
        printf("my-cat: cannot open file\n"); // Error in opening a file
        exit(1);
    }

    char buffer[1024]; // Buffer for the line
    while (fgets(buffer, sizeof(buffer), fp) != NULL) {
        printf("%s", buffer);
    }

    fclose(fp);
    return 0;
}
```

Tämä toimii:

```

markus@markus-X550CL:~/KYSTEEI/Project2$ ./my-cat file.txt
Susikoira Roi oli nuorena innostunut rumpujen soittamisesta, ja soittipa hän ihan bändissäkin. Roin
bändin lupaava nousukiito katkesi kuitenkin Roin päihdeongelmiin ja lopulta kun Roi ei pystynyt
edes omin jaloin kiipeämään esiintymislavalle, bändi päätti laittaa pillit pussiin.

Roi tajusi, että hänen oli ryhdistädyttävä, ja hän pestautui matruusiksi laivalle. Raikas
meri-ilma selvittäisi hänen päänsä ja hän pääsisi aloittamaan elämänsä puhtaalta pöydältä. Vanhat
rumpunsa Roi myi pois yhtä bongorumpua lukuunottamatta. Sitä olisi mukava soittaa laivan kannella
auringonlaskua katsellessa.

Laivalla kaikki sujuikin aluksi hyvin, ja Roi viihdytti muuta henkilökuntaa jokailtaisilla
rumpusooloillaan. Mutta sitten laivalla alkoi levitä kumma tauti. Yhä useampi laivallaolija alkoi
kärsiä ensin merisairauden kaltaisesta pahoinvoinnista, ja sitten vuorokauden kuluessa vaipua
koomaan. Taudin leviämisen hillitsemiseksi sairastuneet vietiin ruumaan perustetulle
karanteeniosastolle.

Roin harmiksi tauti tarttui myös häneen. Roikin vaipui koomaan ja hänet kannettiin ruumaan. Yöllä
tarkastuskierroksella ollut laivan lääkäri ihmetteli Roin huoneesta kantautuvaa meteliä. Lääkäri
kurkisti ovesta olevasta aukosta ja ihmetyksekseen näki Roin hoippuvan ja kaatuilevan ympäri
huonetta. Paikalle saapunut perämies tiedusteli, mitä ihmettä huoneessa tapahtuu. Lääkäri totesi
jopa hieman huvittuneesti: "Harva ruumassa kompuroi, kuin koomassa rumpu-Roi."
markus@markus-X550CL:~/KYSTEEI/Project2$

```

Sitten laajennetaan se lukemaan loopissa kaikki annetut argumentit ja lukemaan kaikki tiedostot. Jos tiedostoa ei anneta, poistutaan palauttaen 0.

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc == 1) { // If no files are given, we just return
        return 0;
    }

    for (int i = 1; i < argc; i++) { // Looping through all files given as arguments
        FILE *fp = fopen(argv[i], "r");
        if (fp == NULL) {
            printf("my-cat: cannot open file\n"); // Error in opening a file
            exit(1);
        }

        char buffer[1024]; // Buffer for the line
        while (fgets(buffer, sizeof(buffer), fp) != NULL) {
            printf("%s", buffer);
        }

        fclose(fp);
    }

    return 0;
}

```

Koodi näyttää toimivan ja tulostaa peräkkäin ensin annetun Susikoira Roi -vitsin ja pätkän Hamletia.

koomaan. Taudin leviämisen hillitsemiseksi sairastuneet vietiin ruumaan perustetulle karanteeniosastolle.

Roin harmiksi tauti tarttui myös häneen. Roikin vaipui koomaan ja hänet kannettiin ruumaan. Yöllä tarkastuskierroksella ollut laivan lääkäri ihmetteli Roin huoneesta kantautuvaa meteliä. Lääkäri kurkisti ovesta olevasta aukosta ja ihmetyksekseen näki Roin hoippuvan ja kaatuilevan ympäri huonetta. Paikalle saapunut perämies tiedusteli, mitä ihmettä huoneessa tapahtuu. Lääkäri totesi jopa hieman huvittuneesti: "Harva ruumassa kompuroi, kuin koomassa rumpu-Roi."

Enter Claudius King of Denmarke, Gertrude the
Queene, Hamlet, Polonius,
Laertes, and his Sister Ophelia,
Lords Attendant.
King.

Though yet of Hamlet our deere Brothers death
The memory be greene: and that it vs befitted
To beare our hearts in greefe, and our whole Kingdome
To be contracted in one brow of woe:
Yet so farre hath Discretion fought with Nature,
That we with wisest sorrow thinke on him,
Together with remembrance of our selues.
Therefore our sometimes Sister, now our Queen,
Th'Imperiall Ioyntresse of this warlike State,
Haue we, as 'twere, with a defeated ioy,
With one Auspicious, and one Dropping eye,
With mirth in Funerall, and with Dirge in Marriage,
In equall Scale weighing Delight and Dole
Taken to Wife; nor haue we heerein barr'd
Your better Wisedomes, which haue freely gone
With this affaire along, for all our Thanks.
Now followes, that you know young Fortinbras,
Holding a weake supposall of our worth;
Or thinking by our late deere Brothers death,
Our State to be disioynt, and out of Frame,
Collegued with the dreame of his Aduantage;
He hath not fayl'd to pester vs with Message,

Ilman argumentteja koodi vain lopettaa ajamisen:

```
markus@markus-X550CL:~/KYTEEMI/Project2$ ./my-cat  
markus@markus-X550CL:~/KYTEEMI/Project2$
```

my-grep.c

Käytetään tämän ominaisuuden tekemisessä string kirjaston funktiota strstr() jonne annettaessa kaksi merkkijonoa, se etsii ensimmäisestä jälkimmäisen substringin ensimmäisen esiintymisindeksin ja palauttaa sen pointerin. Jos pointer on NULL, tiedämme ettei jälkimmäistä stringiä ole ensimmäisessä.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

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

    if (argc == 1) { // If no arguments are given
        printf("my-grep: searchterm [file ...]\n");
        exit(1);
    }

    char *searchstring = argv[1];

    if (strlen(searchstring) == 0) { // If given string is empty, will not print anything
        exit(0);
    }

    // Needed initializations for the getline()
    char *line = NULL;
    size_t len = 0;

    if (argc == 2) { // If no files are given, read from stdin
        while (getline(&line, &len, stdin) != -1) {
            if (strstr(line, searchstring) != NULL) {
                printf("%s", line);
            }
            free(line);
            return 0; // Natural exit
        }
    }

    // Loop through each file specified in the command line arguments
    for (int i = 2; i < argc; i++) {
        FILE *file = fopen(argv[i], "r");
        if (file == NULL) {
            printf("my-grep: cannot open file\n");
            exit(1);
        }

        while (getline(&line, &len, file) != -1) {
            // strstr return a pointer to the first index of a substring (searchstring) so
            // if that is not null, then that string is on the line and we print it.
            if (strstr(line, searchstring) != NULL) {
                printf("%s", line);
            }
        }

        fclose(file);
    }

    free(line);
    return 0;
}

```

Koodin tekemisen välivaiheet valitettavasti hävisivät, kun tekstinkäsittelyohjelma kaatui varoittamatta. Se näyttää nyt kuitenkin toimivan halutulla tavalla.

```

markus@markus-X550CL:~/KYTEEMI/Project2$ ./my-grep thou file.txt file2.txt
And loose your voice. What would'st thou beg Laertes,
What would'st thou have Laertes?
From whence, though willingly I came to Denmarke
My thoughts and wishes bend againe towards France,
markus@markus-X550CL:~/KYTEEMI/Project2$

```

```

markus@markus-X550CL:~/KYTEEMI/Project2$ ./my-grep rumpu file.txt file2.txt
Susikoira Roi oli nuorena innostunut rumpujen soittamisesta, ja soittipa hän ihan bändissäkin. Roin
rumpunsa Roi myi pois yhtä bongorumpua lukuunottamatta. Sitä olisi mukava soittaa laivan kannella
rumpusooloillaan. Mutta sitten laivalla alkoi levitä kumma tauti. Yhä useampi laivallaolija alkoi
jopa hieman huvittuneesti: "Harva ruumassa kompuroi, kuin koomassa rumpu-Roi."
markus@markus-X550CL:~/KYTEEMI/Project2$

```


my-zip.c ja my-unzip.c

Tässä tehtävässä luodaan oma zip- ja unzip. my-zip.c muodostaa RLE-menetelmällä pakatun datan ja tulostaa sen stdoutiin. Data tallennetaan niin, että ensin on 4-tavuinen binääriluku ja sitten merkki, jota se edustaa. 4 tavua on 32 bittiä.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc < 2) { // Check that at least one file is given
        printf("my-zip: file1 [file2 ...]\n");
        return 1;
    }

    for (int i = 1; i < argc; i++) { // Go through all given files
        FILE *file = fopen(argv[i], "r");
        if (file == NULL) {
            printf("my-zip: cannot open file\n");
            return 1;
        }

        int count = 0;
        char current, previous;

        if ((current = fgetc(file)) == EOF) { // Puts in the first value for current or finds out
            // that the file is empty and we can leave this one
            break;
        }
        count++;
        previous = current;

        // Going through the file one character at a time and we increase the counter while it's
        // the same as the previous and go back to one if its not

        while ((current = fgetc(file)) != EOF) {
            if (current == previous) {
                count++;
            } else {
                fwrite(&count, 4, 1, stdout); // Writes a 4-byte integer in binary
                printf("%s", &previous); // Writes the ASCII character
                count = 1; // Resetting counter (next will be the first occurrence)
            }
            previous = current;
        }

        fclose(file);
    }

    return 0;
}
```

Näin ollen käymme jokaisen tiedoston kaikki merkit läpi laskien koko ajan kuinka monta samaa peräkkäistä kirjainta on ja kirjoitamme stdoutiin 4-tavuisen kokonaisluvun binäärillä ja sen perään ASCII merkin.

```
markus@markus-X550CL:~/KYSTEEI/Project2$ gcc -o my-zip my-zip.c -Wall -Werror
markus@markus-X550CL:~/KYSTEEI/Project2$ ./my-zip tobezipped.txt
ah
0b
markus@markus-X550CL:~/KYSTEEI/Project2$ ./my-zip tobezipped.txt > zipped.txt
markus@markus-X550CL:~/KYSTEEI/Project2$ cat zipped.txt
ah
0b
markus@markus-X550CL:~/KYSTEEI/Project2$
```

Tästä on tässä vaiheessa vaikea sanoa, toimiiko se täysin, koska binäärilukuja se ei näytä mukavasti lukevan, mutta se nähdään, kun ollaan tehty unzip.

```

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc < 2) { // Check that at least one file is given
        printf("my-zip: file1 [file2 ...]\n");
        return 1;
    }

    for (int i = 1; i < argc; i++) { // Go through all given files
        FILE *file = fopen(argv[i], "r");
        if (file == NULL) {
            printf("my-unzip: cannot open file\n");
            return 1;
        }

        // Going through the file one character at a time
        int count = 0;
        char ch;
        while (fread(&count, sizeof(int), 1, file) == 1) { // Read the number
            // Read the char and make sure that the file is in correct format
            if (fread(&ch, sizeof(char), 1, file) != 1) {
                printf("my-unzip: invalid file format\n");
                return 1;
            } else {
                for (int i = 0; i < count; i++) { // print as many of the chars as indicated
                    printf("%c", ch);
                }
            }
        }

        fclose(file);
    }

    return 0;
}

```

Tämän jälkeen kuitenkin zipattu tiedosto näyttäisi olevan väärässä formaatissa.

```

markus@markus-X550CL:~/KYTEEMI/Project2$ ./my-unzip zipped.txt
aaaaaaaaaamy-unzip: invalid file format
markus@markus-X550CL:~/KYTEEMI/Project2$ gcc -o my-zip my-zip.c

```

Luultavasti kuitenkin merkki täytyy kirjoittaa zipatessa samalla tavalla kuin numerokin. Tehdään muutos koodiin:

```

while ((current = fgetc(file)) != EOF) {
    if (current == previous) {
        count++;
    } else {
        fwrite(&count, 4, 1, stdout); // Writes a 4-byte integer in binary
        fwrite(&previous, 1, 1, stdout); // Writes the ASCII character
        count = 1; // Resetting counter (next will be the first occurrence)
    }
    previous = current;
}

```

Nyt kaikki näyttää toimivan halutulla tavalla.

```

markus@markus-X550CL:~/KYTEEMI/Project2$ gcc -o my-zip my-zip.c -Wall -Werror
markus@markus-X550CL:~/KYTEEMI/Project2$ ./my-zip tobezipped.txt > zipped.txt
markus@markus-X550CL:~/KYTEEMI/Project2$ ./my-zip tobezipped.txt
a
markus@markus-X550CL:~/KYTEEMI/Project2$ ./my-unzip zipped.txt
aaaaaaaaaabb
markus@markus-X550CL:~/KYTEEMI/Project2$

```

Lisätään vielä koodin loppuun rivinvaihdon tulostus niin ohjelman loppuminen näyttää mukavammalta.

```

markus@markus-X550CL:~/KYTEEMI/Project2$ ./my-unzip zipped.txt
aaaaaaaaaabb
markus@markus-X550CL:~/KYTEEMI/Project2$

```