drives, and rewritable compact disks (CD-RW) are each examples of non-volatile memory.

path A sequence of directory names that specifies the exact location of a file.

text file A file that contains printable characters organized into lines separated by newline characters.

socket One end of a connection allowing one to read and write information to or from another computer.

volatile memory Memory which requires an electrical current to maintain state. The *main memory* or RAM of a computer is volatile. Information stored in RAM is lost when the computer is turned off.

13.11 Exercises

1. Write a program that reads a file and writes out a new file with the lines in reversed order (i.e. the first line in the old file becomes the last one in the new file.)
2. Write a program that reads a file and prints only those lines that contain the substring `snake`.
3. Write a program that reads a text file and produces an output file which is a copy of the file, except the first five columns of each line contain a four digit line number, followed by a space. Start numbering the first line in the output file at 1. Ensure that every line number is formatted to the same width in the output file. Use one of your Python programs as test data for this exercise: your output should be a printed and numbered listing of the Python program.
4. Write a program that undoes the numbering of the previous exercise: it should read a file with numbered lines and produce another file without line numbers.

ch.13_files

October 12, 2020

[1]:
```
#Ch. 13: Files
```

[2]:
```
#create file with "name" and "w" for write.
```

[3]:
```
myfile = open("test.txt", "w")
myfile.write("My first file written from Python\n")
myfile.write("---------------------------------\n")
myfile.write("Hello, world!\n")
myfile.close()
```

[4]:
```
#Read on line at a time
```

[5]:
```
mynewhandle = open("test.txt", "r")
while True: # Keep reading forever
    theline = mynewhandle.readline() # Try to read next line
    if len(theline) == 0: # If there are no more lines
        break # leave the loop

    # Now process the line we've just read
    print(theline, end="")

mynewhandle.close()
```

```
My first file written from Python
---------------------------------
Hello, world!
```

[24]:
```
#Read all lines and return a list of strings
#text file not available
```

[6]:
```
# f = open("friends.txt", "r")
# xs = f.readlines()
# f.close()

# xs.sort()
#
# g = open("sortedfriends.txt", "w")
# for u in xs:
```

```
#     g.write(v)
# g.close()
```

#A good example of a “Filter”-Programm on page 183

#Ch. 13: Exercises

#Ex. 1

[10]:
```
#write file

myfile = open("test.txt", "w")
myfile.write("My first file written from Python\n")
myfile.write("---------------------------------\n")
myfile.write("Hello, world!\n")
myfile.close()
```

[25]:
```
#write filter program

def filter(oldfile, newfile):
    infile = open(oldfile, "r")
    outfile = open(newfile, "w")
    text = infile.readlines()
    text.reverse()       #reverse is modifier
    for everyline in text:
        outfile.write(everyline)
    infile.close()
    outfile.close()

filter("test.txt", "upper.txt")
```

[12]:
```
#write print program

def drucken(name):
    mynewhandle = open(name, "r")
    while True:
        theline = mynewhandle.readline()
        if len(theline) == 0:
            break

        # Now process the line we've just read
        print(theline, end="")

    mynewhandle.close()
```

[13]:
```
#use print program
```

```
drucken("test.txt")
drucken("upper.txt")
```

```
My first file written from Python
---------------------------------
Hello, world!
Hello, world!
---------------------------------
My first file written from Python
```

#Ex. 2

[16]:
```
#write file

myfile = open("test.txt", "w")
myfile.write("dog, [snake, dog], cat\n")
myfile.write("cat\n")
myfile.write("dog\n")
myfile.write("snake\n")
myfile.close()
```

#write filter program

[18]:
```
def read_sub_snake(file):
    infile = open(file, "r")
    while True:
        theline = infile.readline()
        if len(theline) == 0: # If there are no more lines
            break # leave the loop
        if "snake" in theline:
            print(theline)
    infile.close()

#read_sub_snake("test.txt")
```

#Ex. 3

[19]:
```
#write file

myfile = open("list.txt", "w")
myfile.write("dog, [snake, dog], cat\n")
myfile.write("cat\n")
myfile.write("dog\n")
myfile.write("snake\n")
myfile.close()
```

#write addnumber program, which also prints the outputfile

[20]:
```
def addnumber(oldfile, newfile):
    infile = open(oldfile, "r")
    outfile = open(newfile, "w")
    text = infile.readlines()
    count = 1
    for line in text:
        if count < 6:
            outfile.write(str(count) + " ")
        count += 1
        outfile.write(line)
    infile.close()
    outfile.close()
    outfile = open(newfile, "r")
    newtext = outfile.readlines()
    for i in newtext:
        print(i)
    outfile.close()
    return newtext
```

[21]:
```
addnumber("list.txt", "numbered_list.txt")
```

```
1 dog, [snake, dog], cat

2 cat

3 dog

4 snake
```

[21]:
```
['1 dog, [snake, dog], cat\n', '2 cat\n', '3 dog\n', '4 snake\n']
```

#Ex. 4

[22]:
```
#write remove_number program (with print file statement inside)

def remove_number(oldfile, newfile):
    """Write a program that undoes the numbering of the previous exercise: it␣
 ↪should read a file
    with numbered lines and produce another file without line numbers."""
    infile = open(oldfile, "r")
    outfile = open(newfile, "w")
    text = infile.readlines()
    #print(text)
    for line in text:
        outfile.write(line[1:])
    infile.close()
    outfile.close()
```

```
    outfile = open(newfile, "r")
    newtext = outfile.readlines()
    for i in newtext:
        print(i)
    outfile.close()
    return newtext
```

[23]:
```
remove_number("numbered_list.txt", "unnumbered_list.txt")
```

```
dog, [snake, dog], cat

cat

dog

snake
```

[23]:
```
[' dog, [snake, dog], cat\n', ' cat\n', ' dog\n', ' snake\n']
```