Dr. B. Güsmann VGU

Electrical Engineering and Information Technology, B.Eng.
Introduction to the C Programming Language: Exercises

Exercise Sheet 13

1. A pointer to a pointer is e.g. defined by

```
char **PointerToPointer;
```

A pointer to a pointer can be usefull to return a pointer in a function parameter.

```
#include <stdlib.h>

void NeedMemory(char **PointerToPointer){
     *PointerToPointer = (char *)malloc(20);
}

int main()
{
char *NewMemory;

     NeedMemory(&NewMemory);
     return 0;
}
```

Extend main() to write a text of at most 20 characters into the allocated memory. Printf() the text using NewMemory and free the memory.

2. Implement E.8-1 as program copy.c .

a) Call `>copy` in a console window, enter some text as input and watch the echo.

b) Call `>copy > newfile` in a console window.
Enter some text and look with an editor into newfile in your directory.

c) Call `>copy < newfile` in a console window.
Copy is now reading from newfile and printing to the screen.

d) Call `>copy <newfile > secondfile` in a console window.
Look with an editor into secondfile.

3. a) Open the file NewAccess on the harddisk in the write mode and write some text into the file using putc(). Look with an editor into NewAccess. Read the text back using getc() and print it to the screen. Be carefull: you opened the file in the write mode. What do you have to do to read the text out?

b) Open again the now existing file NewAccess in the write mode and write some other text into it. Look with an editor into NewAccess. The old contents should have been disappeared.

c) Open the now existing file NewAccess in the append mode and write another text into the file. Look with an editor into NewAccess. The old contents form exercise part b) should still be there.

4. Using a function of the library.

a) Write a program, that reads 4 int numbers as a char string with %s from the screen. Convert the strings with *atoi()* to int numbers and printf the result with %d to the screen.

b) Write a program, that reads 4 char strings and compares the strings with *strcmp()*. Print the result to the screen.

c) Write a program which reads 1 char string and searches with *strchr()* a specific character within the string. Print the result to the screen.