Das Ziel dieses Praktikums ist die Erarbeitung von Wissen über den System Call `execve`, dessen Funktionsweise und seine Nutzung in C und x86-64 Assembler.

Der Ausgangspunkt für das Praktikum ist das folgende C-Programm:

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

void main() {
    char *binary = "/bin/sh";

    char *arg0 = "/bin/sh";
    char *arg1 = "-c";
    char *arg2 = "ls -al /etc";

    char *argv[] = { arg0, arg1, arg2, NULL };
    char *envp[] =  { NULL };

    int r = execve(binary, argv, NULL);

    perror("execve");
    exit(EXIT_FAILURE);

}
```

Aufgabe 1.

a) Implementieren Sie das C-Programm und führen Sie es aus.

b) Analysieren Sie unter Verwendung des GNU Debuggers die Funktionsweise des Programms.

c) Beschreiben Sie die Funktionsweise des Programms. Beantworten Sie insbesondere folgende Fragen:

1. Wie ist das Speicherlayout des Arrays `argv[]`? Stellen Sie hierzu die Zeigerstruktur als Diagramm dar.
2. Wie werden die Parameter an die Funktion `execve()` übergeben?
3. Wie ist der Stack Frame aufgebaut, der beim Aufruf der Funktion `execve()` auf dem Stack erzeugt wird?
4. Wie und an welcher Stelle werden die übergebenen Parameter im Speicher abgelegt?

5. Was passiert, wenn in Zeile 12 der Wert `arg0` aus dem Array `argv[]` entfernt wird?

Dokumentieren Sie Ihre Erkenntnisse. Nutzen Sie dabei Screenshots mit den Debugging Ausgaben. Geben Sie auch die benutzten GDB Befehle an.

Aufgabe 2. Implementieren Sie ein Assembler-Programm, welches die Funktionsweise des obigen C-Programms nachbildet. Der Code in den Zeilen 17 und 18 muss dabei nicht berücksichtigt werden. Nutzen Sie das in der Vorlesung durch genommene Hello-World Assembler Programm als Grundlage für Ihre Implementierung. Kommentieren Sie das von Ihnen entwickelte Programm.