

Grundlagen - Betriebssysteme und Systemsoftware

IN0009, WiSe 2019/20

Hausaufgabe 0

23.10.2019 23:59 Uhr über Artemis (<https://artemis.ase.in.tum.de>)

Aufgabe 1 Einführung

Keine Abgabe für diese Aufgabe!

a) Allgemeines

- Für die Abgaben und automatisierten Tests nutzen wir Artemis.
- Hausaufgaben sind in Einzelarbeit abzugeben.
- Am Ende des Semesters wird ein Plagiats-Checker alle Abgaben prüfen.
- Falls Sie ein Plagiat abgeben, werden Sie vom Bonussystem ausgeschlossen!
- Die Veröffentlichung und Weitergabe von zur Verfügung gestelltem Material in jeglicher Form (online wie offline) ist ausdrücklich **nicht** gestattet. Verstöße werden als Urheberrechtsverletzung betrachtet und entsprechend geahndet.

b) Artemis

Um an den Hausaufgaben teilzunehmen, müssen Sie sich in Artemis für den Kurs „*Grundlagen Betriebssysteme 2019/20*“ anmelden. Gehen Sie hierzu wie folgt vor:

- Besuchen Sie: <https://artemis.ase.in.tum.de>
- Melden Sie sich mit Ihrem TUM Account (Shibboleth) an.
- Klicken Sie am oberen rechten Rand auf: [Sign up for a course](#)
- Wählen Sie den Kurs „*Grundlagen Betriebssysteme 2019/20*“ aus und melden Sie sich dort an.
- Fertig! Nun können Sie an den Hausaufgaben teilnehmen.

c) Angriffe auf die GBS Infrastruktur bzw. Korruption des Korrektursystems

- Wie in jedem System kann es nicht ausgeschlossen werden, dass die Infrastruktur bzw. unser Korrektursystem Bugs enthält, die zu einer Abweichung vom regulären Betrieb führen. Sollten Sie auf einen solchen Bug stoßen, melden Sie diesen bitte umgehend auf Moodle im **Forum: Detailfragen zu Programmieraufgaben**.
- Sollten Sie einen Exploit gefunden haben, so wenden Sie sich ebenfalls unmittelbar an uns. Sollten Sie sich innerhalb von 12 Stunden nach erfolgreicher Anwendung des Exploits bei uns melden, honorieren wir dies mit **voller** Punktzahl für dieses Aufgabenblatt (Voraussetzung: Sie sind der erste der sich bzgl. Ihres Exploits meldet und Sie beschreiben detailliert, wie dieser durchgeführt wird, damit die Lücke von uns geschlossen werden kann). Sollten Sie sich nicht innerhalb der Frist melden und der Angriff wird bemerkt, gehen wir von böswilligem Handeln aus und Sie werden vom Bonussystem ausgeschlossen.

d) Linux

Für die Bearbeitung der Hausaufgaben wird Linux benötigt. Falls Sie momentan kein Linux auf ihrem System installiert haben, haben Sie folgende Möglichkeiten:

- Installieren Sie Linux (Debian, Fedora, Ubuntu, ...) auf Ihrem Computer (z.B. Dual Boot). **Empfohlen!**

- Nutzen Sie die Rechnerhalle für die Bearbeitung der Aufgaben. Hierzu können Sie sich per SSH und/oder SFTP auf die Rechnerhalle (`lxhalle.in.tum.de`) verbinden. Alle benötigten Programme für die Hausaufgaben sind bereits auf der Rechnerhalle installiert.
- Sie können ebenfalls die von uns bereitgestellte VM nutzen. Hierzu müssen Sie VirtualBox auf Ihrem Computer installieren. Die VM finden Sie unter: <https://gbs.cm.in.tum.de/gbsvm.ova>

Falls Sie Ihr eigenes Linux nutzen wollen, benötigen Sie die folgende Programme:

- git - Quellcodeverwaltung/Abgabe Ihrer Lösungen
- make - Automatisiertes Bauen Ihrer Programme.
- gcc - Unser C Compiler. Artemis nutzt gcc-8.
- libubsan - Falls Sie Ihr Programm mit Undefined Behavior Sanitizer kompilieren wollen (`make ubsan`).
- libasan - Falls Sie Ihr Programm mit Adress Sanitizer kompilieren wollen (`make asan`).
- liblsan - Falls Sie Ihr Programm mit Leak Sanitizer kompilieren wollen (`make lsan`).

```
# Debian based:  
$ apt install git gcc-8 make libasan5 libubsan0 liblsan0  
  
# Red Hat based:  
$ dnf install git gcc make libasan libubsan liblsan
```

e) Rechnerhalle

Wie bekomme ich eine Rechnerhallenkennung?

Wenn Sie sich bis zum **16.10.2019 23:59** in TUMOnline zur Vorlesung angemeldet haben:

Dann müssen Sie einfach Ihre Kennung und Passwort beim Infopoint abholen.

Wenn Sie sich **nicht** angemeldet haben:

Dann müssen Sie am Infopoint eine Rechnerkennung beantragen. Mit diesem Antrag müssen Sie dann zur Übungsleitung gehen und ihn unterschreiben lassen. Danach kann der Infopoint Ihnen eine Kennung erstellen. Die Übungsleitung finden Sie in Zimmer 00.05.038 - Martin Uhl. Eventuell kurz vorher via Telefon ankündigen (089 289 18088).

Wenn Sie Ihre Kennung oder Passwort vergessen haben:

Der Helpdesk, welchen Sie in der kleinen Rechnerhalle 00.07.023 finden, ist Ihnen behilflich diese Daten wieder zu finden oder Ihr Passwort neu zu setzen. Hierzu solltet Ihr einen offiziellen Lichtbildausweis mitbringen.

Kurzanleitung

In der Rechnerhalle einfach einen Rechner durch Tastendruck einschalten, Linux bestätigen und sich dann mittels der Informatik Rechnerkennung anmelden.

Remote

Man kann sich mittels des SSH Protokolls weltweit mit dem Rechner (`lxhalle.in.tum.de`) verbinden, welcher dieselbe Umgebung wie die Rechner in der Rechnerhallen besitzt. Weiteres finden Sie unter: <https://wiki.in.tum.de/Informatik/Helpdesk/Ssh>

Das RDP-Protokoll bietet von überall grafischen Remote-Zugriff auf die Rechnerhalle (`lxhalle.in.tum.de`). Näheres finden Sie unter: <https://wiki.rbg.tum.de/Informatik/Benutzerwiki/RemoteDesktop>

Aufgabe 2 Bash

In dieser Aufgabe sollen Sie sich mit dem Terminal, hier der GNU Bash, vertraut machen. Nutzen Sie bitte das zur Verfügung gestellten *Linux Onepage* und eine Suchmaschine Ihrer Wahl, um sich das nötige Wissen anzueignen. Grundlegende Kenntnisse der GNU Bash sind zur Bearbeitung aller weiteren Hausaufgaben nötig.

Sie haben verschiedene Möglichkeiten, eine GNU Bash aufzurufen:

- Sie loggen sich über SSH auf der Rechnerhalle ein und nutzen die dort zur Verfügung gestellte Infrastruktur (empfohlene Variante).
- Wenn Sie eine native Linux-Distribution auf Ihrem Computer installiert haben, können Sie die Bash direkt über die Liste aller Applikationen öffnen.
- Sie nutzen eine online-Bash, wie z.B:
https://www.tutorialspoint.com/execute_bash_online.php.
Solche ist gut geeignet um einzelne Befehle auszuprobieren. Ob eine online-Bash allerdings genügt, um diese Aufgabe zu bearbeiten, hängt von der verwendeten Bash ab.
- Sie nutzen das Windows Subsystem for Linux. Installationsanweisungen finden Sie im Internet.
Vorsicht: Die WSL ist aktuell in seinem Funktionsumfang nicht identisch mit einer nativen Linuxumgebung. Wenngleich Sie diese Aufgabe mit dem WSL bearbeiten können, so werden nicht alle Hausaufgaben damit ausführbar sein.
- Sie nutzen eine Linux VM. Dies ist z.B. mit VirtualBox und dem von uns bereitgestellten Debian Image möglich.

Um diese Aufgabe möglichst realitätsnah zu gestalten, geben wir Ihnen die Ordnerstruktur und die Dateien eines fiktiven Softwareprojekts vor. Sie erhalten in Ihrem Git Repository folgende Dateien:

- project.tar.gz
- setup.sh
- script.sh

Bitte bearbeiten Sie die folgenden Aufgaben. Um Ihnen den permanenten Wechsel zwischen Aufgabenblatt und Lösungsdatei zu ersparen, steht die komplette Aufgabenstellung nochmals als Kommentar in der Dateivorlage. Diese Kommentare werden vom Tester nicht berücksichtigt und können von Ihnen beliebig bearbeitet bzw. gelöscht werden.

Sie können ein Skript ausführen, indem Sie innerhalb des Ordners, welcher das Skript enthält, Folgendes eingeben:

```
./<Dateiname des Skripts>
```

Es kann sein, dass dies nicht funktioniert, da dieses Skript nicht als „ausführbar“ gekennzeichnet ist. Um es als ausführbar zu markieren, geben Sie Folgendes ein:

```
chmod +x <Dateiname des Skripts>
```

Aufgabenstellung

1. Fügen Sie ein „Shebang“ in die erste Zeile beider Skriptdateien ein. Gehen Sie davon aus, dass die Bash unter dem Standardpfad installiert wurde.

2. Das Skript `setup.sh` dient dazu, die Ordnerstruktur anzulegen, auf der `script.sh` arbeiten soll. Durch Ausführen von `setup.sh` können Sie diese Ordnerstruktur insbesondere zurücksetzen, sodass Sie `script.sh` ausführlich testen können. Nach dem eingefügten Shebang soll der Ordner `project` komplett gelöscht werden (unabhängig davon ob dieser existiert). Anschließend entpacken Sie mit Hilfe des Befehls `tar` das vorgegebene tar-Archiv. Nach Ausführung des Skripts `setup.sh` sollten die folgende Dateien/Ordner im Git Repository enthalten sein

- `project`
- `project.tar.gz`
- `setup.sh`
- `script.sh`

Die restlichen Aufgaben beziehen sich auf `script.sh`.

Wechseln Sie in den Ordner `project`

3. Geben Sie **sämtliche** Inhalte (also auch versteckte Dateien) des aktuellen Ordners als Liste auf `stdout` aus.

Wechseln Sie in den Unterordner `doc`

4. Dieser Ordner enthält die Dokumentation des Projekts, welche automatisch erstellt werden kann (vgl. Doxygen, Sphinx, ...). Die aktuell enthaltene Dokumentation ist allerdings veraltet, daher sollen **alle** Inhalte des aktuellen Verzeichnisses (außer `.` und `..`) gelöscht werden.

Wechseln Sie zurück in den Ordner `project`

5. Der Name des Ordners `INCLUDE` entspricht nicht der Konvention des Projekts nach der alle Ordnernamen klein geschrieben werden sollen. Benennen Sie diesen Ordner in `include` um.

6. In Softwareprojekten werden oft gewisse Codekonstrukte verboten, da diese als veraltet oder fehleranfällig gelten. Sie wollen sicher gehen, dass dieses Projekt nirgendwo die Funktion `assert (....)` nutzt. Listen Sie alle Dateien auf, die das Wort `assert` enthalten (nur den Dateinamen).

7. Sie wollen nun eine neue Version ihrer Software veröffentlichen. Daher müssen alle Versionsnummer in der Datei `Readme.md` durch die neue Versionsnummer ersetzt werden. Die aktuelle Version ist 3.15.1. Die neue Versionsnummer ist 3.16. Ersetzen Sie alle Vorkommen der alten Versionsnummer durch die neue Versionsnummer.

8. Softwareprojekte können durch verschiedene Metriken klassifiziert werden. Eine sehr simple Metrik ist die Anzahl der Codezeilen (LOC). Geben Sie die Anzahl der Zeilen (nicht nur Codezeilen) im Ordner `src` aus. Ihre Ausgabe muss das folgende Format haben:

`LOC: <Anzahl> total`

9. Wiederholen Sie Aufgabe 8, aber schreiben Sie die Ausgabe im identischen Format in eine Datei `LOC.txt`
10. Beenden Sie ihr Skript mit dem Rückgabewert 2.

Abzugebende Dateien:

- `setup.sh`
- `script.sh`

Hinweis:

- Sorgen Sie dafür, dass die Ausgabe Ihres Programms exakt mit den Anforderungen übereinstimmen. Achten Sie insbesondere auch auf nicht sichtbare Steuerzeichen, wie Leerzeichen und Zeilenumbrüche.

- Bedenken Sie stets, dass diese Aufgabe eine einführende Aufgabe ist. Entsprechend kurz und simpel können die einzelnen Lösungen sein. Sollten Ihre Lösungen sehr lang werden, so gibt es vermutlich eine einfacheren Weg.
- Die Dateigröße von `setup.sh` und `script.sh` darf jeweils **1MB** nicht überschreiten.
- Jedes Skript muss innerhalb von **10 Sekunden** terminieren
- Der Tester prüft Ihr Programm mit nicht deterministischen Eingabedaten