

Allgemeine Hinweise zu den SP-Übungen

- Die Aufgaben sind teils alleine, teils in Gruppen zu bearbeiten (siehe Aufgabenstellung). Bei Gruppenarbeit sind Lösungsweg und Programmierung gemeinsam zu erarbeiten.
- Zur Versionsverwaltung wird `git` eingesetzt. Dafür muss ein an das IDM gekoppelter Account auf <https://gitlab.cs.fau.de> erstellt werden.
- Für jede Aufgabe ist die Bearbeitungszeit in Werktagen (bei uns nur Montag bis Freitag) angegeben. Die Bearbeitungszeit beinhaltet den Tag der eigenen Tafelübung. Nicht enthalten sind vorlesungsfreie Tage und Feiertage. Mit Hilfe des Kommandos `/proj/i4sp1/bin/deadline <aufgabeX>` kann der effektive Abgabetermin für den Aufrufenden angezeigt werden.
- Die Übungsaufgaben müssen spätestens bis zum jeweiligen Abgabetermin abgegeben werden. In darauffolgenden Tafelübungen werden einzelne abgegebene Lösungen besprochen – jeder Übungsteilnehmer muss dabei in der Lage sein, die gesamte Lösung seiner Gruppe zu erläutern. Kann jemand seine Lösung auf Anforderung nicht erklären, wird für ihn die Aufgabe als nicht abgegeben bewertet (im Zweifelsfall kann hierzu ein Gespräch außerhalb der Tafelübung stattfinden).
- Die abgegebenen Programme werden automatisch auf Ähnlichkeit mit anderen Programmen desselben Semesters und früherer Semester überprüft. Werden starke Übereinstimmungen festgestellt, wird die Aufgabe als nicht abgegeben bewertet.
- Kann ein Übungstermin nicht wahrgenommen werden, kann vorher mit dem eigenen Übungsleiter ein Ersatztermin in einer anderen Übungsgruppe vereinbart werden.
- Durch ausführen des Skriptes `/proj/i4sp1/bin/mkrepo` kann für jede Aufgabe ein neues Projekt-Repository im Gitlab erzeugt werden. Dieses wird automatisch mit den Vorgabedateien befüllt.
- Der letzte Stand (am Hauptzweig **main**), der zum Abgabezeitpunkt in dem Gitlab-Repository (`git push`) vorgefunden wird, wird als Abgabe gewertet.
- Sollten Sie aus triftigem Grund eine fristgerechte Abgabe versäumen, so ist eine verspätete Abgabe durch Aufruf des Abgabeprogramms möglich. Damit eine solche Abgabe bei der Wertung berücksichtigt wird, ist jedoch in jedem Fall eine Rücksprache mit Ihrem Übungsleiter erforderlich. Eine vorhandene rechtzeitige Abgabe wird durch eine verspätete Abgabe nicht gelöscht.

Aufgabe 0: epoche (1.0 Punkte)

Die UNIX-Epoche definiert einen fixen Zeitpunkt, der als Ausgangspunkt für Datumsberechnungen in UNIX-kompatiblen Betriebssystemen dient. Im Detail bezieht man sich damit auf den 1. Januar 1970, 00:00 Uhr UTC. Ihre Aufgabe ist es nun, ein Program zu schreiben, das die verstrichenen Sekunden seit der "UNIX-Epoche" ausgibt.

Erstellen Sie sich zunächst ein neues Repository mit `mkrepo` und clonen Sie dieses anschließend, sodass Sie eine lokale Arbeitskopie haben (siehe <https://docs.gitlab.com/topics/git/clone/>). In dieser können Sie dann eine Datei `epoche.c` anlegen, die ihr Programm enthalten soll. Erstellen Sie eine leere `main`-Funktion mit einer entsprechenden Signatur (`int main(void)`).

Ihr Code sollte nun bereits mit `gcc` übersetzbar sein. Das resultierende Programm sollten Sie ausführen können, auch wenn noch keine Ausgabe sichtbar ist:

```
$ gcc -std=c11 -pedantic -Wall -Werror -D_XOPEN_SOURCE=700 \
    -fsanitize=undefined -g epoche.c -o epoche
$ ./epoche
```

Nun können Sie die eigentliche Funktion implementieren. Verwenden Sie dazu die Bibliotheksfunktionen **`time(3p)`** und **`printf(3p)`**. Mehr Informationen zu diesen Funktionen finden Sie in den Man-Pages, z.B:

```
$ man 3p time
```

Die Ausgabe soll am Ende zum Beispiel so aussehen:

```
$ ./epoche
Sekunden seit der UNIX-Epoche: 1738106803
```

Um die Aufgabe abzugeben müssen Sie ihre lokale Kopie mit `git add` und `git commit` registrieren und mit `git push` hochladen (siehe <https://docs.gitlab.com/topics/git/commit/>).

Hinweise zur Abgabe:

Bearbeitung: Einzeln

Bearbeitungszeit: 3 Werktage (ohne Wochenenden und Feiertage)

Abgabezeit: 17:30 Uhr