



FAU • Dekanat der TF
Martensstraße 5a
91058 Erlangen

FAU • Dekanat der TF • Martensstr. 5a • 91058 Erlangen

Herr
M. Sc. Bernhard Heinloth
(PERSÖNLICH)

WS'19/20: Auswertung zu Übungen zu Betriebssysteme

Sehr geehrter Herr M. Sc. Heinloth,

Sie erhalten hier die Ergebnisse der automatisierten Auswertung der Lehrveranstaltungsevaluation im WS'19/20 zu Ihrer Umfrage vom Typ "Übung":

- Übungen zu Betriebssysteme -

Es wurde hierbei der Fragebogen - t_w19u20 - verwendet, es wurden 22 Fragebögen von Studierenden ausgefüllt.

Der Wert 1 kennzeichnet hierbei eine maximale Güte, der Wert 5 eine minimale Güte für die einzelnen Fragen bzw. Mittelwerte.

Der Kapitel-Indikator für "3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter" zeigt den mit der Anzahl der Antworten gewichteten Mittelwert der 6 Hauptfragen und damit den Lehrqualitätsindex (LQI), dieser wird bei genügend (ab 5) Rückläufern zur Qualitätssicherung durch die Studienkommissionen und auch für die Bestenlisten der verschiedenen Kategorien verwendet.

Der Kapitel-Indikator für "5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter" zeigt den Mittelwert für die restlichen Einzelfragen, diese dienen nur der Information der Dozentin/des Dozenten.

Bei den Einzelfragen werden je nach Fragen-Typ die Anzahl und Verteilung der Antworten, Mittelwert und Standardabweichung aufgelistet.

Die Text-Antworten für jede offene Frage sind zusammengefasst aufgelistet.

Eine Profillinie zeigt den Vergleich zu den Mittelwerten aller Rückläufer für diesen Fragebogen-Typ. Die Profillinie eignet sich auch zur Präsentation in der LV.

Eine Einordnung Ihrer Bewertung ist nach Abschluss der Ergebnisauswertung unter

<https://eva.tf.fau.de> --> Ergebnisse --> WS'19/20 möglich, siehe Bestenlisten, Percentile, etc.

Bitte melden Sie an tf-evaluation@fau.de die Anzahl der ausgegebenen TANn, wenn Sie das bis jetzt versäumt haben.

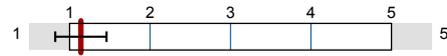
Mit freundlichen Grüßen

Rolf Wanka (Studiendekan, rolf.wanka@fau.de)
Jürgen Frickel (Evaluationskoordinator, tf-evaluation@fau.de)



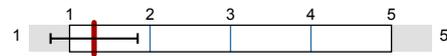
Globalwerte

3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter



mw=1,14
s=0,32

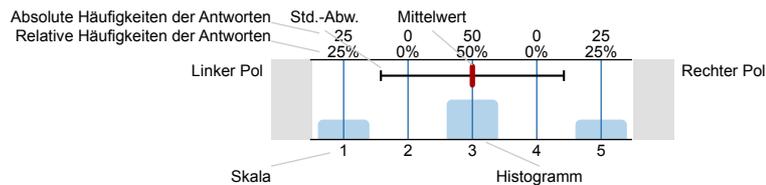
5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter



mw=1,31
s=0,54

Legende

Fragetext



n=Anzahl
mw=Mittelwert
s=Std.-Abw.
E.=Enthaltung

2. Allgemeines zur Person und zur Lehrveranstaltung

2.1) ▶▶ Ich studiere folgenden Studiengang:

INF • Informatik	<input type="text" value="21"/>	21	n=22
IuK • Informations- und Kommunikationstechnik	<input type="text" value="1"/>	1	

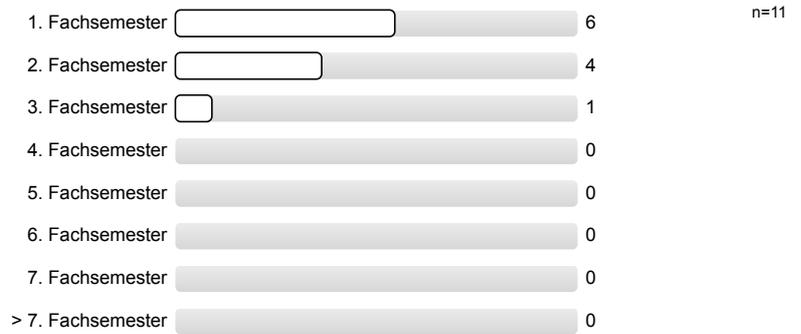
2.2) ▶▶ Ich mache folgenden Abschluss:

B.Sc. • Bachelor of Science	<input type="text" value="11"/>	11	n=22
M.Sc. • Master of Science	<input type="text" value="11"/>	11	
M.Sc.(hons) • Master of Science with Honours	<input type="text" value="0"/>	0	
M.Ed. • Master of Education	<input type="text" value="0"/>	0	
LA • Lehramt mit Staatsexamen	<input type="text" value="0"/>	0	
Dr.-Ing. • Promotion	<input type="text" value="0"/>	0	
Zwei-Fach-Bachelor of Arts	<input type="text" value="0"/>	0	
Sonstiges	<input type="text" value="0"/>	0	

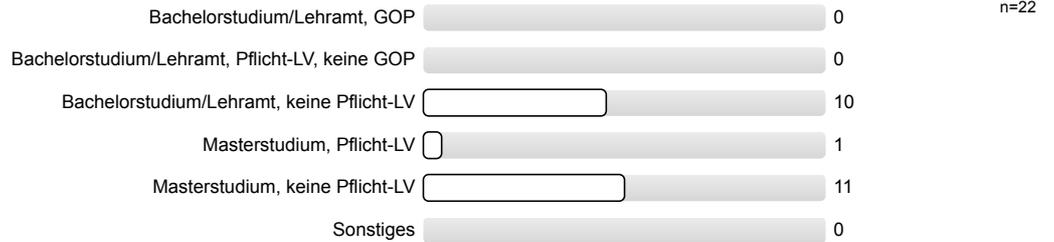
2.3) Ich bin im folgenden Fachsemester (im Bachelor):

1. Fachsemester	<input type="text" value="0"/>	0	n=11
2. Fachsemester	<input type="text" value="0"/>	0	
3. Fachsemester	<input type="text" value="0"/>	0	
4. Fachsemester	<input type="text" value="0"/>	0	
5. Fachsemester	<input type="text" value="9"/>	9	
6. Fachsemester	<input type="text" value="0"/>	0	
7. Fachsemester	<input type="text" value="2"/>	2	
8. Fachsemester	<input type="text" value="0"/>	0	
9. Fachsemester	<input type="text" value="0"/>	0	
> 9. Fachsemester	<input type="text" value="0"/>	0	

2.4) Ich bin im folgenden Fachsemester (im Master):



2.5) ►► Diese Lehrveranstaltung gehört für mich zum

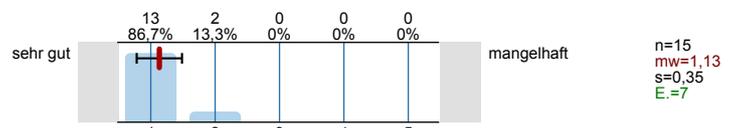


2.7) Ich besuche etwa Prozent dieser Übung.

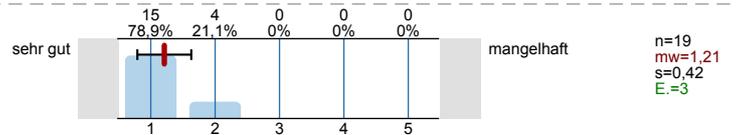


3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter

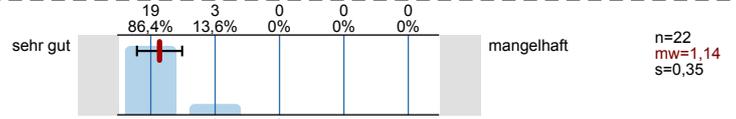
3.1) ►► Die Übung entspricht den im Modulhandbuch eingetragenen Inhalten und Kompetenzen.



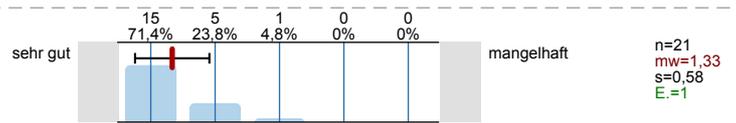
3.2) ►► Wie ist die Einpassung in den Studienverlauf Ihres Studienganges?



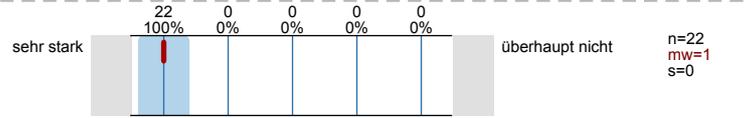
3.3) ►► Wie ist die Übung selbst strukturiert?



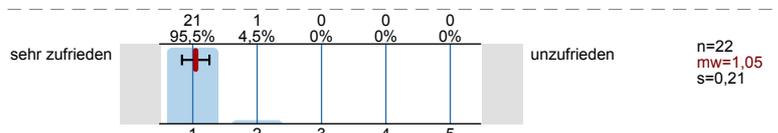
3.4) ►► Wie ist die Übung inhaltlich und organisatorisch mit der zugehörigen Vorlesung abgestimmt?



3.5) ►► Die Übungsleiterin/Der Übungsleiter wirkt engagiert und motiviert bei der Durchführung der Übung.



3.6) ►► Wie zufrieden sind Sie insgesamt mit der Übung:



4. Kommentare zu Lehrveranstaltung und Übungsleiterin/Übungsleiter

4.1) An der Lehrveranstaltung gefallen mir folgende Aspekte besonders:

- * Inkrementelles Erreichen eines Zieles über das ganze Semester
- * Echtes x86_64, kein obsukrer "Akademischer Assembler" oder Handychip-geraffel.
- * Rechnerübungsfragen werden super beantwortet :)
- * C++ ist eine super Sprache *duck und weg*
- - Lockerer Umgangston und Bestärkung durch die ÜbungsleiterInnen, Fragen zu stellen und Fehler zu machen
- Ausreichende Anzahl an ÜbungsleiterInnen, um bei Problemen schnell Hilfe zu erhalten
- - Sehr klarer Anwendungsbezug
- genau richtiger Schwierigkeitsgrad/Zeitaufwand pro Übung
- Seminar; auch wenn nicht direkt notwendig für BS (z.B. Bootseminar), sehr interessant
- entspanntes Übungsumfeld
- - Sehr motivierte Übungsleiter
- Man lernt wirklich extrem viel und es macht Spaß
- Entspannte Atmosphäre (sowohl in der Rechnerübung, als auch in der Übung)
- Das Bearbeiten der Übungsaufgaben wird einem nicht unnötig erschwert: Durch Vorgabe der Makefiles (kvm, gdb, netboot) und die Testhardware (+Remote-testen von Zuhause) muss man kaum Zeit für unnötige Dinge (wie Debugger einrichten, zum Testen in die Uni fahren, etc.) aufwenden und kann sich direkt auf die eigentlichen Übungsaufgaben konzentrieren -> wenig verschwendete Zeit
- Sehr gut ausgewählte Übungsaufgaben, sehr hoher Lerneffekt
- schöne, übersichtliche Folien
- - Übungsleiter sind alle sehr motiviert
- Fragen werden erklärt. Also wirklich erklärt und nicht nur die Lösung erzählt
- Aufbauende Aufgaben, nicht zu viel Vorgegeben, aber auch wenig "Schreibarbeit", 64bit funktioniert schön
- BS war der letzte Schritt im Studium, um die Funktionsweise von der nackten Hardware bis zum laufenden Programm zumindest im Ansatz zu durchdringen.
- BS war die beste Übung, die ich im Studium bis jetzt hatte. Großes Lob an die Organisation und das Engagement!
- Beispiele schrittweise durchgehen war gut (Stack-Aufbau beim context_switch z.B.)
- Bestes Fach. Macht mega Spaß und es wird in den Rechnerübungen auch immer sehr viel erklärt.
- Die tatsächlicher Programmier-Erfahrung, die wir hier vermittelt bekommen. Das wird unter anderem dadurch massiv verstärkt, dass sich die Übungsleiter bei den Abgaben den gesamten Code ansehen und nicht nur auf Fehler überprüfen und dann auch Verbesserungsvorschläge machen, die den Code besser machen.
- Die Übung ist top.
- Die Übung vertieft den Stoff der Vorlesung so gut, dass man die Konzepte bis in die Tiefe verstehen kann. Obwohl mitunter nicht ganz leicht, ist bei mir nie der Punkt der Verzweiflung erreicht worden. Bei Fragen helfen engagierte Tutoren schnell und freundlich. Alles in Allem mein Lieblingsfach bisher.
- Herangehensweise, Vortragsart, Eingang auf Fragen, Unstieg auf 64 Bit Systeme
- Man merkt das sehr viel Herz in die Veranstaltung gesteckt worden ist. Zudem macht es auch Spaß, endlich mal ein ernsthaftes Projekt zu programmieren.
- Sehr lehrreiche Abnahmen. Man macht wirklich was in der Vorlesung besprochen wird. Sehr gute Hilfestellungen und doch wird nichts verraten.
- alle tutoren und Übungsleiter sind (oder wirken überzeugend) enthusiastisch.

4.2) An der Lehrveranstaltung gefällt mir Folgendes weniger, und ich schlage zur Verbesserung vor:

- - Bernhard sagt nicht oft genug historisch gewachsen
- - Rechnerübung auch am Wochenanfang (Mo/Di) anbieten ==> bitte nicht zur 1. Stunde
- vereinzelt waren Folien zur Tafelübung im Vorfeld nicht online verfügbar ==> ist schade für die, die mitschreiben möchten
- - Übung Donnerstag und Freitag ist bloed (besser ueber die Woche verteilen waere schoen)
- C++ hat mich am Anfang echt verwirrt, vor allem bei der 1. Aufgabe. Wenn man das Konzept der Stream Ausgabe nicht kennt, verwirrt das am Anfang ziemlich, was überhaupt zu machen ist. Aber danach ging es bergauf und die späteren Aufgaben fielen mir um einiges leichter.
- Die Uhrzeiten. Ich hab leider bei jeder Übung einen Konflikt mit zB Cloud Computing, weshalb ich nicht so häufig in der Übung sein kann.
- Dispatcher und Scheduler sind immer noch Instanzklassen ;) (im Gegensatz zu dem Rest, was alles Namespaces sind)

- Es ist ein wenig schade, dass die freiwilligen Zusatzaufgaben immer aufeinander aufbauen. Hat man in der ersten Aufgabe die Zusatzaufgabe nicht mit gelöst, so fällt das für das ganze Semester (zumindest bis jetzt) weg. Besser wäre es hier, wenn die Zusatzaufgaben nach Bearbeitungszeit gelöst würden, so dass man in der nächsten Abgabe direkt weiter machen kann.
- Ich stehe nicht so auf Dominosteine vor Weihnachten :-)) lieber mehr von diesen Sandwich Spekulatius
- Mehr STL-Bibliothek:
Es mag sowohl technische als auch ideologische Gründe geben, wieso wir überhaupt keine Standardbibliothek haben, aber das sorgt immer wieder für unnötigen Zusatzaufwand. Ich sehe keinen Mehrwert darin, ein schlechteres `sprintf` zu bauen (ich denke an `std::enable_if`, `O_stream`), die `std::enable_if` Implementierung aus der STL kopieren zu müssen (für `Queue / QueueLink` Interaktion), etc. Natürlich gibt es vernünftige Gründe, warum die ganze STL nicht funktionieren kann (man denke an all die Syscalls, und alles was damit verknüpft ist wie z.B. `cout`). Vielleicht gibt es - gegen C++ Ideologie, das man sich in den Fuß schießen darf wenn man das will - auch Dinge die nicht funktionieren sollten (`malloc / free` Lecks im BS fixen könnte sehr grausam werden, ohne all die Tools die natürlich `cout / cerr` brauchen), auch wenn man dabei unnötig "sichere" Klassen mit vernichtet (`std::vector`, ich meine dich.)

Aber `std::enable_if_t`, `std::is_same`, `std::numeric_limits`, `std::accumulate`, `sprintf` uvm. sind alle keine dieser Kategorien. Es wäre schön sie einfach verwenden zu können. Es sollte afaik ja C++-Runtimes für Firmware-Entwicklung geben, die das gleiche Problem haben und deren Standardbibliothek könnte man vermutlich relativ schmerzlos integrieren.

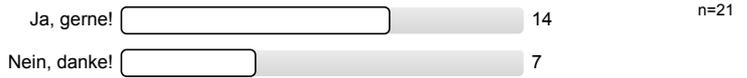
Niemand sollte dazu gezwungen werden "modernes, ideomatisches C++" zu schreiben - nicht Fokus der VL - aber dazu gezwungen zu werden alle erlernten Idome aus dem Fenster zu werfen mit "Du bekommst keine STL" ...
- Teilweise der Vorlesung zu weit voraus.
Folien sind schwierig als Referenz (viele sind 5 mal hintereinander wegen den Einblendungen).
- C++ muss nicht gleich auf einem Schlag eingeführt werden, mit allen Vor- und Nachteilen. ggf wäre eine stufenweise Einführung -- bei Bedarf/nach Übung -- denkbar. auch nett wäre eine Art Sammlung von Fehler-Meldungen, welches das Bearbeiten der ersten paar Aufgaben wesentlich einfacher machen würde.

4.3) Zur Lehrveranstaltung möchte ich im Übrigen anmerken:

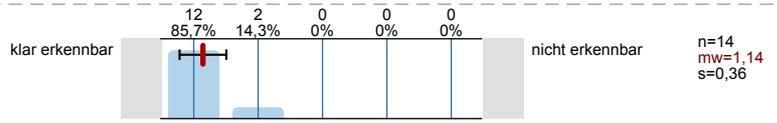
- - C++ Einführung war relativ knapp und nur mäßig gut dokumentiert. Das hätte etwas ausführlicher sein können
- • **unsachlicher Kommentar gelöscht** • (vom Evaluationskoordinator der TF)
- Alles super Leute!
- Apfelpunsch war sehr lecker
- Die Idee mit dem Seminaren finde ich sehr gut auch wenn es nur 3 Stück waren (gerne mehr).
Das einzige was mich etwas stört ist, dass man trotz allem das Gefühl hat, als würde man stets noch an der Oberfläche kratzen. Was aller Dings verständlich ist da es schwierig ist in einem 5/7,5 ECTS Modul, ein so flächendeckendes Thema abzudecken. Hilft also nur sich im Master darin zu vertiefen :)
- Es ist manchmal schade nicht in die Zukunft schauen zu können. Aufgabe 0 geht super mit `sprintf` - und dann kommt Aufgabe 1 und plötzlich gibt es kein `sprintf` mehr, in Aufgabe 1 ist es legitim polling auf der Tastatur zu machen (Leitet ja zu Aufgabe 2), aber genauso gibt es andere Dinge die man nicht tun sollte (Aufgabe 3: voller Epilog unter `BKL.lock()` ist nicht sinnvoll (Wo ist der Unterschied zu Aufgabe 2?), aber Epilog ganz ohne `BKL.lock()` wird dann in der Besprechung auch als potentielles Problem aufgeführt, weil sich der Kontext in dem man das Verwenden soll mit Aufgabe 5 ändern wird)
Es ist verständlich dass man von Korrektur her nur schaffen kann indem man jeden mehr oder weniger auf den selben Weg zerrt - keine SoftIRQ Implementierung statt Prolog-Epilog - aber im Gegensatz zu den Tutoren haben die Studenten nicht im Kopf was in dieser Aufgabe noch "akzeptabel" ist weil in späteren Aufgaben "aufgebessert", und was "zu viel" ist, weil es in späteren Aufgaben Probleme verursachen wird oder spätere Aufgaben "alternativ" - aka unkorrigierbar - löst.
- Es wäre wirklich sehr schön wenn mehr Veranstaltungen vom Lehrstuhl 4 nicht in der dritten Säule wären. Ich würde gerne noch VM oder BST hören, komme aber leider schon vorher auf 30 ECTS in der Säule (IT Sec & Lehrstuhl 3 & VS & CC).
- Ich habe im Studium bisher wenige Doktranden gesehen, die sich in das Beiwerk Lehre so stark reinhängen wie Bernhard.
- In der Übung wurden Beispiele genannt bezüglich Dingen, auf die wir hier eingehen sollen. Die konnte ich mir aber nicht merken, es wäre vllt. hilfreich, die irgendwo auf der Webseite dazuzuschreiben :)
- Manchmal musste man etwas lange auf einen Tutor warten. Insgesamt lief es aber super.
- Super motiviertes, sympathisches, engagiertes und lockeres Team!!! Weiter so!
- Vielleicht die Folien/ Doku noch fertig zu Ende ändern (Bsp. A4 man braucht bei Context doch noch ne Wrapper Funktion, Folien manchmal noch im Ablauf für die alte Aufgabe aufgebaut --> macht's später schwieriger zu folgen, Bsp. Context Wechsel Asm: mussten die richtigen Register ewig suchen, weil die Tabelle ganz oben ist und die Sachen in 'umsetzung in mpstubs' haben dafür weniger hergegeben)
bevor man es veröffentlicht.
Zu dem Aufbau via enums: finde ich gut;
Via namespaces: ich persönlich würde Klassen bevorzugen, weil ich C++ doch gerne ein wenig üben möchte und da Klassen (mit deren Funktionen bspw. Konstruktor) häufiger verwendet als selbstgeschriebene Namespaces

5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter

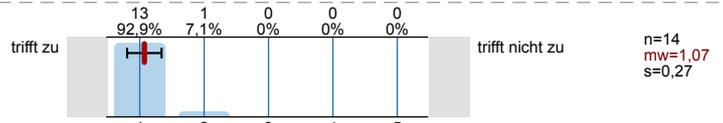
5.1) Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter beantworten?



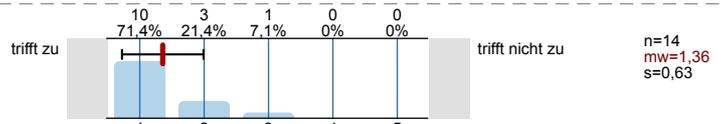
5.2) Zielsetzungen und Schwerpunkte des Übungsinhalts sind:



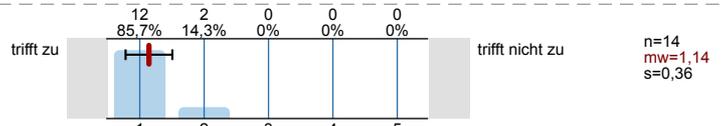
5.3) Ich werde gut zum selbstständigen Lösen von Aufgaben angeleitet.



5.4) Die Anwendbarkeit des Übungsstoffes wird z.B. durch Beispiele gut verdeutlicht.



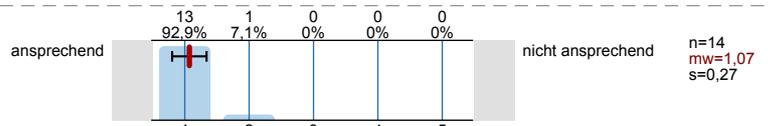
5.5) Die Übungsform (Aufgabenbehandlung, Programmieren, etc.) ist gut zur Vermittlung des Stoffes geeignet.



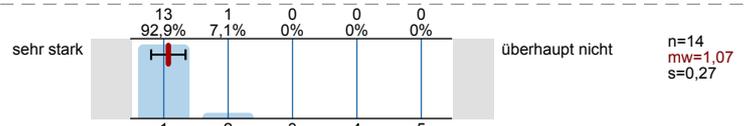
5.6) Die Präsentation von Aufgaben und Lösungen ist nachvollziehbar, es ist genügend Zeit zum Mitdenken vorhanden.



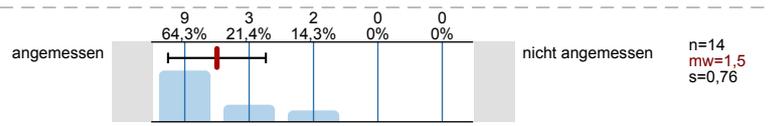
5.7) Der Präsentationsstil der Übungsleiterin/des Übungsleiters ist:



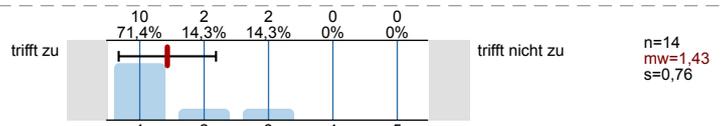
5.8) Die Übungsleiterin/Der Übungsleiter geht auf Fragen und Belange der Studierenden ein.



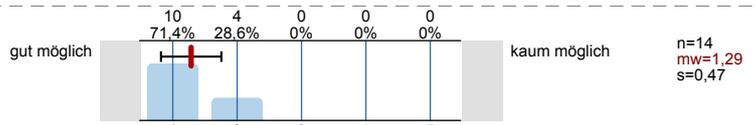
5.9) Der Einsatz und das Zusammenspiel von Medien (Tafel, Overhead-Projektor, Beamer, etc.) ist:



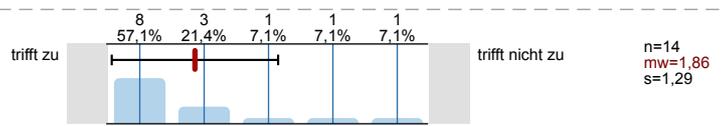
5.10) Die zur Verfügung gestellten Unterlagen sind in Menge und Qualität den Zielen der Übung angemessen.



5.11) Anhand des erarbeiteten Übungsmaterials ist die Vertiefung des Vorlesungs-/Modulinhalts:

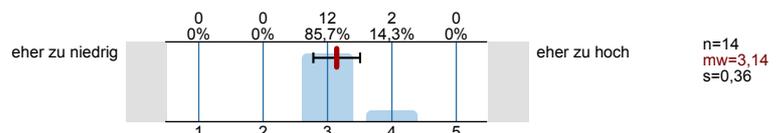


5.12) Der Bezug zu den Prüfungsanforderungen wird hergestellt.

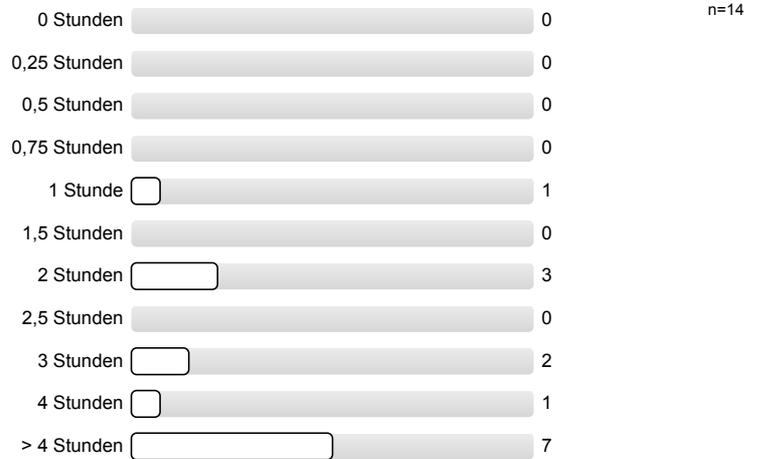


6. Schwierigkeitsgrad und Aufwand

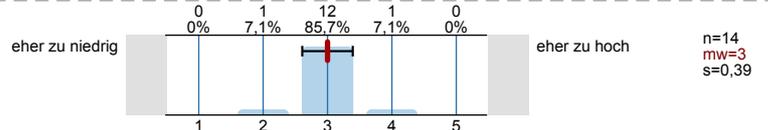
6.1) Der Schwierigkeitsgrad der Übung ist:



6.2) Mein Durchschnittsaufwand für Vor- und Nachbereitung dieser Übung beträgt pro Woche:

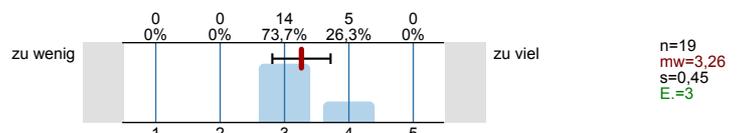


6.3) Meinen zeitlichen Durchschnittsaufwand für diese Übung finde ich:



7. Von der Übungsleiterin/Vom Übungsleiter gestellte Fragen

7.1) Wie empfanden Sie den Umfang des von uns vorgegebenen Quellcodes?

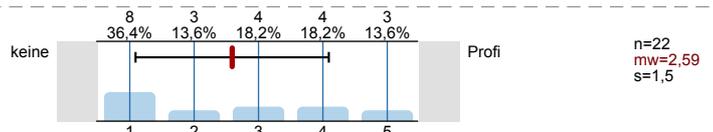


7.2) Weitere Anmerkungen zum vorgegebenen Quellcode und zur Dokumentation (Verständlichkeit, Umfang unserer Erläuterungen, ...)

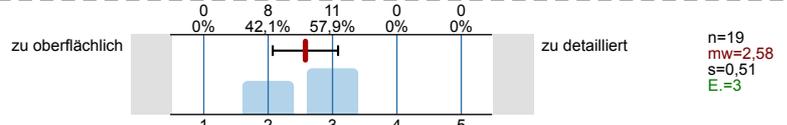
- - Gute Hilfestellung mit Bitfeldern, vorimplementierte Module nützlich
- Umfang ok, man braucht sich nicht durch massig Doku/Code hangeln, um die Aufgaben anzugehen
- - einrückung war nicht konsistent, hat sauber-machungs commits benötigt (so was wie editorconfig wäre echt zu empfehlen)
- es ist nicht ganz einfach sich zu merken welche klassen so gespeichert sind (unsicher wie das ohne recursive-such-tools verbessert werden kann)
- - manche Beziehungen zwischen Klassen waren nur durch das Klassendiagramm ersichtlich (nicht durch Beschreibungen). Kurzer Hinweis in der Uebung dann fragt man sich weniger was der Zweck mancher vorgaben ist (stellvertretendes Beispiel die Klasse Secure)
- Die Dokumentation war insgesamt hilfreich, auch wenn sie an manchen wenigen Stellen noch nicht auf das aktuelle Semester angepasst war. Das hat manchmal verwirrt.
- Dokumentation war super (auch wenns auf deutsch etwas komisch zu lesen ist). Es wär cool gewesen wenn man eine ganze Klasse selber schreiben könnte (z.B. Watch oder Secure). Dafür evtl eine der Zusatzaufgaben weglassen. OStream::operator<< hätte ein Template sein können :(
- Es sind immer mal Fehler oder Ungenauigkeiten vorhanden, bei Hinweis auf solche reagieren die Übungsleiter allerdings positiv und beheben das Problem möglicherweise gleich.
- Evtl. etwas mehr Assembler selber schreiben
- Ich glaube ich habe immer noch keine Ahnung welche Implementierung in welchem Ordner ist.
- Im Verlauf des Semesters wird die Anzahl an Klassen mit Unterklassen/Structs/Enums enorm. Auch die Anzahl an Quellcodedateien ist riesig, sodass der Überblick über das Projekt hintenraus nicht mehr leicht zu behalten ist. Evtl. könnten Abschnitte/Klassen, die für die letzten Aufgaben kaum benötigt werden, reduziert werden?
- Siehe oben

- Verständlicher Code und sehr gute Kommentare, allerdings passiert trotzdem recht viel Magie durch Klassen welche man selbst nie anfasst.
- Wesentlich besser als bei ungenannten anderen Veranstaltungen

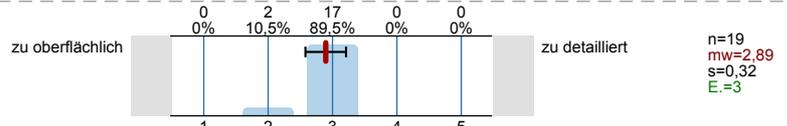
7.3) Welche Vorkenntnisse in C++ waren vorhanden?



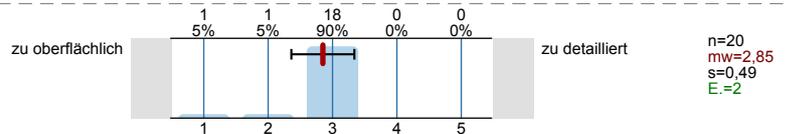
7.4) Die Einführung in C++ war...



7.5) Die Einführung in GDB war...



7.6) Die Einführung in Assembler war...



7.7) Weitere Anmerkungen zu den freiwilligen Zusatzseminaren

- - Bootloaderseminar fand ich persönlich sehr interessant
- Assembler war praktisch.
- Beibehalten, ist interessant
- Beim Bootseminar könnte man noch die Frage kurz anreißen, ob andere Architekturen wie RISC-V die Sache besser machen.
- Die GDB-Einführung war hilfreich.
- Die waren mit das Beste an der Übung. Ich hab zwar nicht alles verstanden, aber es war sehr interessant, mal über den Tellerrand zu schauen und mehr zu hören, als das was man implementieren soll. Auch das historisch gewachsene (:) war sehr interessant.
- Ein tolles Format. Ist interessant und macht Spaß.
- Ich konnte leider nicht alle besuchen
- Kann mich nicht ganz erinnern aber in c++ vllt aufgaben spezifisch bisschen drauf eingehen (namespaces, enums etc)
- Konnte zeitlich nicht teilnehmen
- Tolle Sache, die Inhalte konnte man alle brauchen bzw. waren interessant fuer das Gesamtverstaendnis unseres Programms.
- Urlader ist esoterisch aber unterhaltend.
Der Rest nicht so wirklich freiwillig in meiner Wahrnehmung - mehr so wie die TÜ selber auch: Wer das Thema schon kann kann gerne zu Hause bleiben, aber wer das Thema nicht kennt und nicht kommt hat es vermutlich wirklich massiv schwierig.
- Waren Interessant und haben zum Bearbeiten der Übungsaufgaben geholfen (spätestens beim debuggen)
- ich glaube im gdb seminar hätte man ehr auf methoden des debuggings setzen sollen, als eine aufzählung von befehlen (nicht das das keinen sinn macht).

7.8) Anmerkungen speziell zur Rechnerübung (Ablauf, Anzahl, Qualität der Betreuung, Test-Hardware, ...)

- - einmal erklären wie man von aussen auf die Testhardware zugreift. Dann kann das auch jeder :)
- 4-Stunden Slots sehr gut.
Rechnerübungen könnten in der Woche ein bisschen verteilter liegen (nicht nur am Ende der Woche). Falls man Fragen hat, müsste man dann nicht so lange warten.
- Betreuung sehr gut :)
- Die Qualität der Betreuung war super, ich fand es toll, dass neben den eigentlichen Tutoren die Übungsleiter auch immer anwesend waren.
Auch schön, dass mit dem zusätzlichen Termin am Mittwoch um 8 auf Überschneidungen eingegangen wurde!
Und dass man das ganze System von zu Hause testen kann, macht es sehr angenehm, danke dafür. Ich finde es immer störend,

wenn man nur in der Uni die Abgaben bearbeiten kann. Das wurde toll gelöst.

- Die Rechnerübung finde ich fuer BS essentiell wichtig (teilweise doch recht technische Probleme zu deren Loesung und Verstaendnis man Hilfe braucht), die Betreuer nehmen sich zum Glueck viel Zeit zum Debuggen, Erklaeren und Helfen, das ist Klasse und traegt zur Gesamtqualitaet des Moduls entscheidend bei.
Leider ist die Mannschaft aber scheinbar etwas unterbesetzt, so dass es gerade dadurch, dass die Leute sich fuer jeden einzelnen Studenten Zeit nehmen, haeufig zu langen Wartezeiten kommt und die Zeit in den Rechneruebungen teilweise gar nicht ausreicht (die Betreuer bleiben dann manchmal sogar noch laenger um zu helfen!).
Ausserdem wurden nur 2 Rechneruebungen pro Woche (bzw spaeter noch eine 3. Uebung Mittwochs um 8Uhr) angeboten, wenn man da keine Zeit hat ist das echt bloed, auch deshalb koennte man vllt. etwas mehr Personal brauchen?
Sonst echt super!
- Die anderen Tutoren und Doktorranden koennen Bernhard nicht das Wasser reichen und sind fuer meinen Geschmack nicht ausfuehrlich und gruendlich genug. Sorry.
- Hardware war super (immer ein Rechner frei, VNC ist geil)
Übungsslots waren schwierig gelegen aber sehr lang, also konnte man immer irgendwann unterbringen.
Man ist auch halbwegs schnell drangekommen.
- Ich wuensche mir, dass alle Tutoren versuchen, genau wie Bernhard unsere Loesungen bei der Abgabe bis ins Detail auseinanderzunehmen. ;-)
- Netboot ist super!

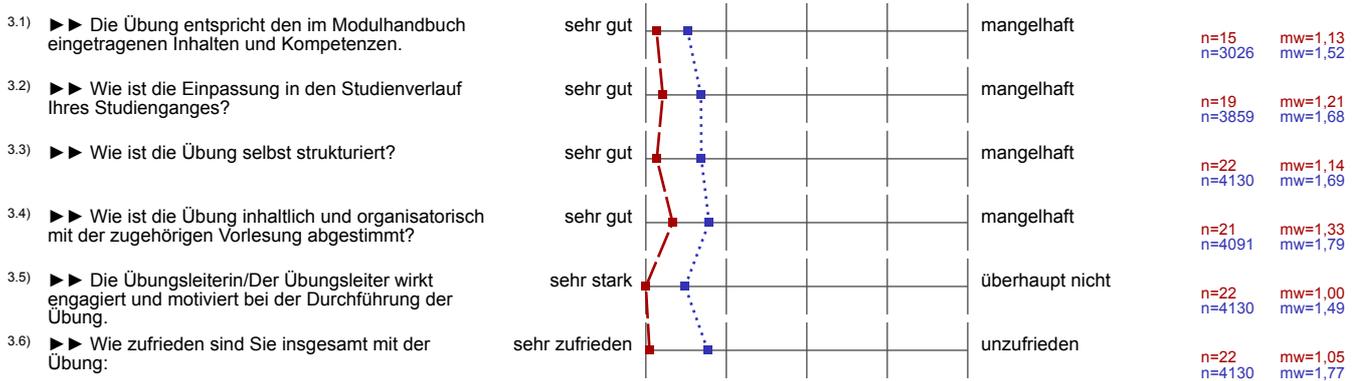
Ich bin eigentlich kein RÜ-Student, vielleicht Repos wie in SP / anderen LV so eirichten (lassen), dass auch Studentische Fragen per Mail sinnvoll beantwortet koennen, weil Einblick auf den Code fuer Tutoren gegeben ist?
- Passt alles
- Sehr gute Infos und Tipps bei der Abgabe
- Top
- war immer ganz angenehm.

Profilinie

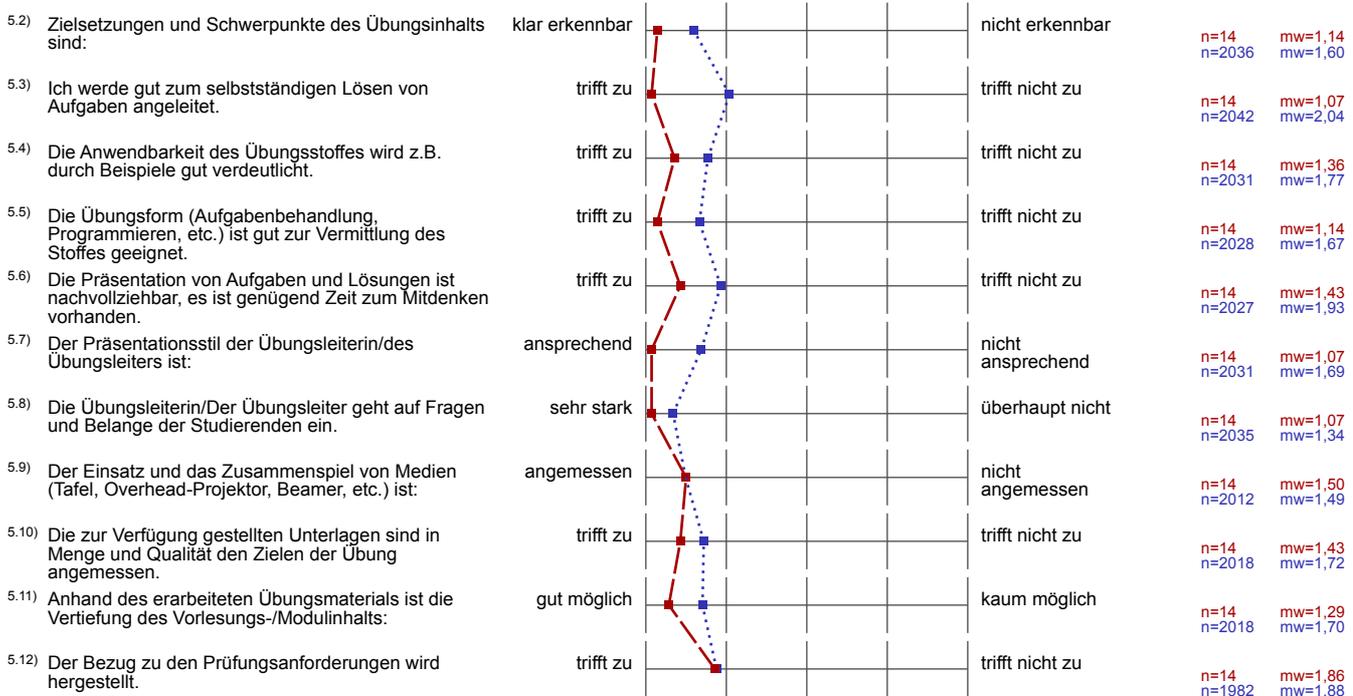
Teilbereich: Technische Fakultät (TF)
 Name der/des Lehrenden: M. Sc. Bernhard Heinloth
 Titel der Lehrveranstaltung: Übungen zu Betriebssysteme (Name der Umfrage)
 Vergleichslinie: Mittelwert aller Übungs-Fragebögen WS'19/20

Verwendete Werte in der Profillinie: Mittelwert

3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter



5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter



6. Schwierigkeitsgrad und Aufwand



