

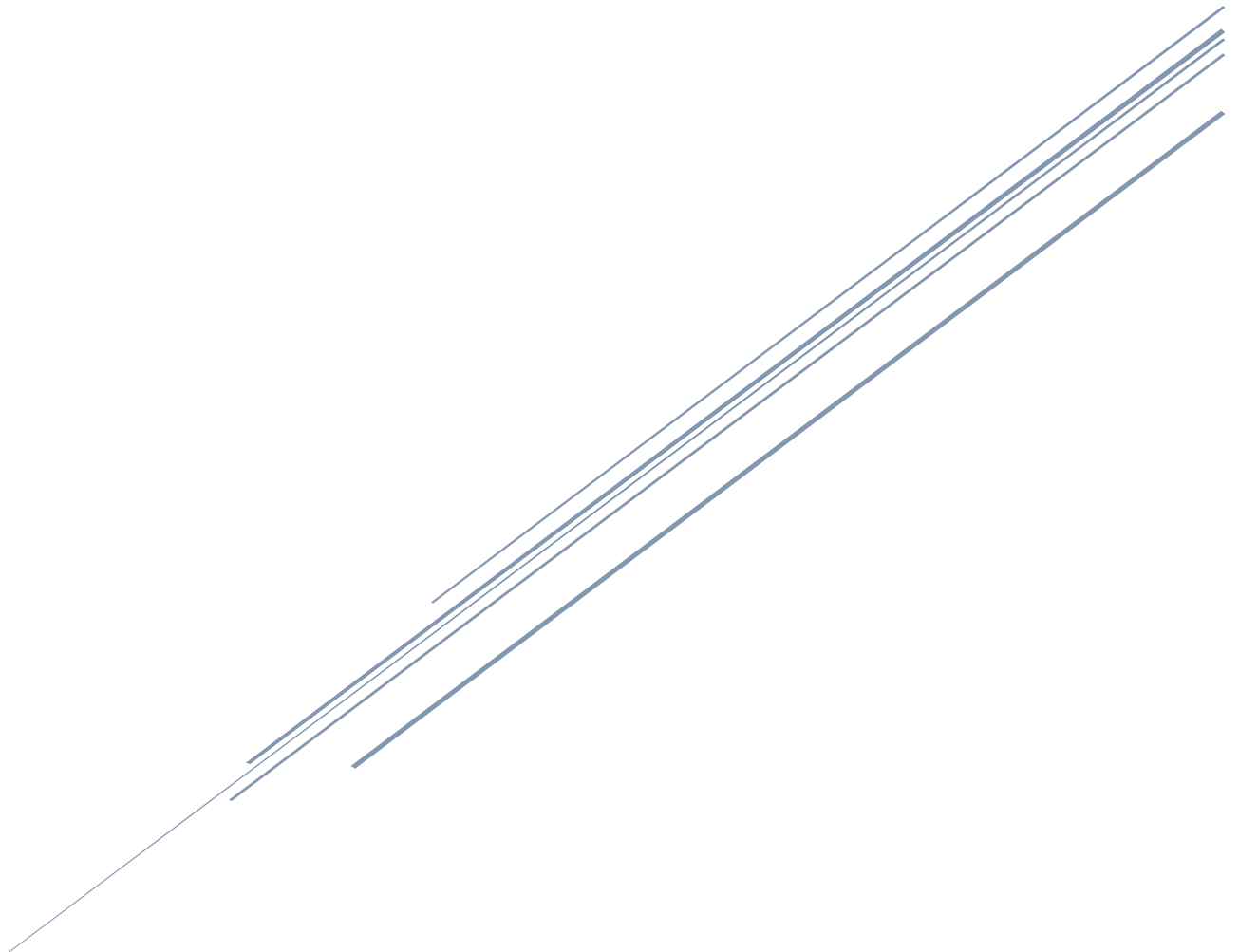
MEDIANFILTER

Projektleiterbericht Assembler

Praktikum: Einführung in die Rechnerarchitektur

Sommersemester 2018

Gruppe 33



Gruppenmitglieder:

- Projektleiter: Ruben Bachmann (Matrikel-Nr.: 03693902)
- Dokumentation: Franziska Steinle (Matrikel-Nr.: 03702739)
- Präsentation: Roland Würsching (Matrikel-Nr.: 03648533)

Eingereicht am: 08.07.2018

Inhalt

Entstehung der Lösung:.....	2
Bewertung des Projekterfolgs:.....	3
Projektverlauf und Umsetzung des Zeitplans:	5
Spezifikation:	5
Implementierung:.....	5
Vergleich von erwarteter und tatsächlicher Dauer:.....	6
Aufgabenverteilung:.....	6
Entscheidungen:	7
Positive und negative Ereignisse:	8
Positive Ereignisse:	8
Negative Ereignisse / Probleme:	8

Entstehung der Lösung:

Die Entstehung unserer Lösung begann im Wesentlichen mit dem ersten Gruppentreffen. Zu diesem Zeitpunkt hatte sich jedes Gruppenmitglied die Aufgabenstellung bereits einmal durchgelesen, es waren jedoch noch einige Fragen bezüglich des genauen Ziels der Aufgabe, sowie möglicher Lösungen offen.

So musste unter Anderem geklärt werden, auf welchen Daten wir genau arbeiten müssten, wie und wo diese im Speicher zu finden sein würden, welche Funktion das C-Rahmenprogramm hätte, sowie welche Funktionalitäten beziehungsweise Unterprogramme wir für unsere Lösung benötigen würden. Zwar wurden all diese Fragen bereits im groben während des Treffens geklärt, jedoch entschied unsere Gruppe, dass es vermutlich das Beste wäre, wenn sich jedes Gruppenmitglied eine Woche lang eigenständig Gedanken zu möglichen Lösungen machen würde, sodass wir bei unserem zweiten Treffen unterschiedliche Ansätze zur Verfügung hätten.

Beim zweiten Gruppentreffen exakt eine Woche darauf haben wir dann unsere selbst erarbeiteten Lösungen zusammengetragen, welche alle im Wesentlichen den gleichen Grundgedanken hatten. So kamen wir zu dem Schluss, dass es sinnvoll wäre die Aufgabe in vier Teilaufgaben zu unterteilen. Diese Teilaufgaben waren bzw. sind wie bereits in der Spezifikation beschrieben: Kopieren, Sortieren, Tauschen sowie die eigentliche Hauptfunktion, welche die drei anderen miteinander verbindet.

Während dieses Treffens wurde ebenfalls diskutiert, wie man diese Funktionen umsetzen und in Assemblercode implementieren könnte. Die Kopier- Tausch- sowie die Hauptfunktion waren schnell geklärt, da diese relativ simpel und nur auf wenige Arten sinnvoll implementiert werden konnten. Die Implementierungsmöglichkeiten der Sortierfunktion sowie die Wahl eines passenden Sortieralgorithmus gestalteten sich hingegen etwas schwieriger. Zur Lösung dieses Teilproblems stellte jedes Gruppenmitglied die ihm bekannten Sortieralgorithmen vor, und es wurde kurz diskutiert ob, und wenn ja wie diese mit überschaubarem Aufwand in Assemblercode implementiert werden könnte. Zusätzlich zu den uns bereits bekannten Sortieralgorithmen haben wir uns im Internet über weitere Sortierverfahren informiert, sind jedoch zu dem Schluss gekommen, dass alle uns noch unbekannten Verfahren nur schwer in Assembler zu implementieren wären.

Letzten Endes blieben zwei vom Prinzip her relativ ähnliche Sortierverfahren zur Auswahl übrig: Selectionsort und Bubblesort, welche zwar nicht die schnellsten Sortieralgorithmen sind, jedoch eine für unsere Aufgabe akzeptable Laufzeit besitzen, und sich durch ihre Einfachheit auszeichnen. Da dies das einzige Unterprogramm war, für welches wir zwei unterschiedliche Lösungsansätze finden konnten, entschloss sich unsere Gruppe für die Spezifikation Selectionsort und Bubblesort als je einen Lösungsansatz zu präsentieren. Auf die letztendliche Entscheidung welchen der Algorithmen wir für unsere Implementierung verwenden würden, wird unter ‚Entscheidungen‘ genauer eingegangen. Damit war der Rahmen für unsere Lösung gesetzt, und die Spezifikation konnte geschrieben werden.

Um sicher zu stellen, dass die Aufgabe richtig verstanden wurde, sowie zur Abklärung letzter kleiner Fragen bezüglich der Spezifikation, vereinbarte unsere Gruppe ein Treffen mit dem Tutor. Aufgrund von Terminüberschneidungen konnten nur zwei Gruppenmitglieder an diesem Treffen teilnehmen, da allerdings nur kleine Fragen zu besprechen waren entschieden wir uns das letzte Gruppenmitglied im nächsten Gruppentreffen über den Verlauf zu informieren.

Da es während des Treffens mit dem Tutor zu keinen Überraschungen kam, konnte die Spezifikation zügig fertiggestellt werden und nach einem abschließenden Treffen zur Besprechung des Treffens mit dem Tutor sowie letzter Formulierungen, pünktlich in den Abgabeordner hochgeladen werden. Während dieses Treffens wurde außerdem bereits besprochen, wie der Zeitliche Ablauf der Implementierungsphase aussehen sollte. Genauerer dazu finden Sie unter ‚Projektverlauf‘.

In unserem ersten Treffen nach Abgabe der Spezifikation, haben wir diese Vorläufige Zeitplanung dann vervollständigt. Des Weiteren nahmen uns vor, bis zu unserem nächsten Treffen die Implementierung größtenteils abzuschließen. Aufgrund der gründlichen Planung während der Spezifikationsphase konnten wir dann auch schnell mit der Implementierung beginnen, sodass der Code tatsächlich bereits bei unserem zweiten Treffen der Implementierungsphase zu großen Teilen fertiggestellt war. Während dieser Zeit tauschten wir uns vor Allem über WhatsApp beziehungsweise Kommentare im Code aus. Zu Beginn arbeiteten wir dabei noch parallel an zwei leicht unterschiedlichen Versionen, stellten jedoch schnell fest, dass eine der Beiden wesentlich übersichtlicher war, und konzentrierten uns fortan nur noch auf diese.

Während des Treffens wurden dann fast nur noch Kleinigkeiten Besprochen, da sowohl der Ablauf, als auch Fragen zum Code bereits in vorigen Treffen oder über WhatsApp geklärt worden waren. Die einzige größere Frage welche aufkam war in Bezug auf die Verwendung von 32 Bit Registern. Da sich die Gruppe nicht sicher war ob und wie diese verwendet werden dürfen, beschlossen wir uns noch ein weiteres Mal mit dem Tutor zu treffen, und bei dieser Gelegenheit auch letzte formale Fragen zu klären.

Auch bei diesem Treffen einigten wir uns darauf, dass es ausreichend wäre, wenn nur zwei Gruppenmitglieder erscheinen, und das letzte Gruppenmitglied beim nächsten Treffen beziehungsweise über WhatsApp über den Verlauf des Treffens informiert wird. Da es wie beim ersten Treffen mit dem Tutor zu keinen wirklichen Überraschungen kam, konnten wir die Implementierung ohne größere Schwierigkeiten beenden, und unsere Implementierung, die erstellten Tests, das Makefile sowie die Readme nach einem abschließenden Treffen, in welchem wir nur noch sicherstellten, dass nichts vergessen worden war, und dass alle Kommentare angemessenen Umfang hatten, pünktlich in den Abgabeordner hochgeladen werden. Hiermit war das Projekt bis auf die Rollenspezifischen Aufgaben erfolgreich beendet.

Bewertung des Projekterfolgs:

Das Ergebnis unserer Implementierung ist insgesamt als Erfolg einzustufen, da alle Erfolgskriterien, welche in unserer Spezifikation unter den Punkten ‚Soll-Analyse‘ beziehungsweise ‚Bemerkungen‘ zu finden sind, vollständig erfüllt wurden. Das erste dieser Kriterien bestand darin, die Zahlenfolge in Blöcke ungerader Größe einzuteilen. Diese Funktion wird innerhalb unserer Sortierfunktion im vollen Umfang bereitgestellt, welche ebenfalls das zweite Erfolgskriterium, nämlich das Sortieren dieser Blöcke, erfüllt.

Das dritte Erfolgskriterium legte fest, dass die gefilterte Zahlenfolge an die Zielspeicherstelle geschrieben werden soll. Dieses Kriterium wird in zwei Schritten erfüllt. Zuerst wird die gesamte Zahlenfolge an die Zielspeicherstelle kopiert, was durch unsere Kopierfunktion geschieht. Nachdem die Blöcke dann sortiert wurden, werden die Mediane der sortierten Blöcke mit Hilfe einer zweiten Funktion in richtiger Reihenfolge an die Zielspeicherstelle geschrieben, wodurch auch das Gesamtziel der Aufgabe, nämlich das filtern der ursprünglichen Zahlenfolge, erfüllt wird.

Die übrigen Erfolgskriterien bestanden darin Tests, ein Makefile sowie eine Readme Datei welche die Bedienung des Programms erklärt, zur Verfügung zu stellen. Auch dies ist uns unserer Einschätzung nach im vollen Maße gelungen.

Über die eigentliche Aufgabenstellung hinaus hatten wir uns in unserer Spezifikation als zusätzliches Erfolgskriterium festgelegt, eine Graphische Ausgabe der ursprünglichen und der sortierten Zahlenfolge zu erstellen. Auch dieses Kriterium konnte mit Hilfe des C-Rahmenprogramms erfüllt werden.

Projektverlauf und Umsetzung des Zeitplans:

Alles in Allem war es immer Ziel der Gruppe anstehende Aufgaben möglichst frühzeitig zu erledigen. Dies hatte zum einen den Vorteil, dass wir kurz vor den Abgabeterminen in keine Stresssituation kamen. Zum anderen war es uns dadurch möglich sowohl bei der Spezifikation, als auch bei der Implementierung die Abgabedateien gründlich zu prüfen, und somit unnötige Fehler zu vermeiden. Dieses Ziel konnten wir eigentlich immer einhalten, was unter anderem auch an den festen Zielen lag, welche wir uns am Ende jedes Treffens für das nächste Treffen gesetzt haben. Im Folgenden wird dieser Zeitplan im groben dargestellt:

Spezifikation:

- Woche 1: Ziel der ersten Woche war es, sich gemeinsam einen Überblick über die Aufgabe zu verschaffen, Fragen zu klären, sowie erste Lösungsansätze zu entwickeln. Dies sollte mit Hilfe eines Gruppentreffens, sowie individueller Überlegungen geschehen.
- Woche 2: Der Plan der zweiten Woche bestand daraus, die in der ersten Woche entwickelten Lösungsansätze zusammenzutragen, und mit der Ausformulierung dieser zu beginnen. Auch dies sollte wieder mit Hilfe eines Gruppentreffens geschehen.
- Woche 3-4: Ziel der letzten beiden Wochen vor der Abgabe der Spezifikation war es, den bereits in der zweiten Woche erstellten Fließtext zu vervollständigen, sowie dessen Formulierungen zu verbessern. Des Weiteren wollten wir bereits mit der Planung der Implementierungsphase beginnen, sodass wir möglichst frühzeitig mit dieser beginnen könnten.

Implementierung:

- Woche 1: Für die erste Woche der Implementierungsphase hatten wir uns vorgenommen, die Planung der Implementierung vollständig abzuschließen, sowie mit der Umsetzung unseres Lösungsansatzes in Code zu beginnen.
- Woche 2: In der zweiten Woche der Implementierungsphase wollten wir bereits die Umsetzung des Lösungsansatzes in Code weitestgehend abschließen, sowie mit der Ausarbeitung erster Tests beginnen. Was und wie genau getestet werden sollte wurde in einem Treffen besprochen.
- Woche 3-4: Ziel für die dritte sowie vierte Woche war es dann, die Implementierung, Tests, Makefile sowie Readme vollständig abzuschließen, sowie alles auf seine korrekte Funktion hin zu überprüfen. In einem abschließenden Treffen sollten dann letzte Fragen geklärt werden.
- Woche 5: In der letzten Woche vor der Abgabe sollten dann nur noch die Kommentare im Code so angepasst werden, dass sie auch für einen Außenstehenden gut verständlich und nachvollziehbar waren.

Die oben erwähnten Treffen fanden immer mittwochs um 12 Uhr statt, da dies wie im ersten Treffen festgestellt wurde, für alle der beste Termin war. Zusammenfassend ist zu sagen, dass dieser Zeitplan sehr ähnlich dem in unserer Spezifikation ist, und da es uns immer gelang diese Ziele einzuhalten, von einem erfolgreichen Projektverlauf die Rede sein kann.

Vergleich von erwarteter und tatsächlicher Dauer:

Das Finden von Lösungsansätzen verlief um einiges schneller als erwartet, da alle Gruppenmitglieder im Wesentlichen die Selbe Vorstellung davon hatten, wie das Projekt implementiert werden sollte. So konnten wir uns schnell auf eine Einteilung des Gesamtproblems in Teilprobleme einigen. Der einzige Aspekt bei dem diese Einigung etwas länger dauerte, war das Festlegen auf einen Sortieralgorithmus, jedoch geschah auch dies in einem annehmbaren Zeitrahmen.

Die Erstellung der Spezifikation hat insgesamt etwas weniger Zeit in Anspruch genommen als erwartet, da unsere Gruppe sehr gut zusammengearbeitet hat. So übernahm jedes Gruppenmitglied einen Teil der Erstellung des Fließtextes oder überarbeitete den vorher von einem anderen Gruppenmitglied erstellten Text. Durch dieses parallele Arbeiten konnte diese Phase somit recht schnell abgeschlossen werden.

Das Erstellen der Tests, des Makefiles sowie der Readme verliefen im Großen und Ganzen etwa so schnell wie erwartet. Da es hier zu keinen wirklichen Überraschungen kam.

Einzig und allein die eigentliche Implementierung dauerte etwas länger als erwartet, da sich das Implementieren als etwas schwieriger herausstellte als erwartet, und wir aufgrund der Frage zur Verwendung von 32 Bit Registern kurzen Leerlauf hatten. Jedoch konnte auch die Implementierung noch innerhalb unseres Zeitplans abgeschlossen werden.

Aufgabenverteilung:

Für die Spezifikation haben wir die Aufgaben relativ gleichmäßig verteilt. Das bedeutet, dass jedes Gruppenmitglied an jedem einzelnen Aspekt (also Lösungsfindung, Diskussion und Ausformulierung), in etwa demselben Umfang mitgearbeitet hat. Dies war auch einer der Gründe dafür, weshalb die Spezifikation etwas schneller als gedacht fertiggestellt werden konnte. Zudem hat sich jedes Gruppenmitglied die Texte, welche von anderen Gruppenmitgliedern geschrieben wurden, noch einmal durchgelesen und eventuelle Fehler und Unklarheiten korrigiert.

Da wir uns dazu entschieden hatten das TUM LaTeX Template für unsere Abgabe zu verwenden, und sich Roland als einziger aus unserer Gruppe mit LaTeX auskannte, erstellte er schließlich die finale Abgabe aus dem vorher gemeinsam bearbeiteten Word Dokument.

Auch das C-Rahmenprogramm von ihm geschrieben, da er sich ebenfalls am besten mit der Programmiersprache C auskannte. Der Rest der Implementierung wurde jedoch in Zusammenarbeit erstellt und getestet.

Um an dieser Stelle nur kurz das gesamte Praktikum anzusprechen (also inklusive VHDL Projekt): Jedes der Gruppenmitglieder hatte seine Stärken in einem anderen Bereich, und hat entsprechend dieser an unterschiedlichen Bereichen und Projekten mehr, an anderen weniger gearbeitet. Insgesamt gab es jedoch eine relativ faire Aufgabenverteilung.

Entscheidungen:

Da es für unsere Aufgabe nur wenige Möglichkeiten gab die Lösung auf unterschiedliche Art und Weise anzugehen beziehungsweise zu implementieren, kam es im Wesentlichen nur zu zwei großen Entscheidungen. Die erste dieser Entscheidungen war die Wahl des Sortieralgorithmus, bei welcher wir uns zwischen Selectionsort und Bubblesort. Unsere Gruppe hat sich letztendlich wie bereits in der Spezifikation beschrieben für den Selectionsort Algorithmus entschieden, da dieser für unsere Zwecke eine geringfügig bessere Laufzeit hat, sowie etwas einfacher zu implementieren ist.

Wie bereits erwähnt hätte man als alternativen Lösungsansatz auch einen komplizierteren Algorithmus wie Mergesort oder Quicksort verwenden können, dies wäre jedoch unserer Meinung nach nicht praktikabel gewesen.

Die zweite große Entscheidung bestand darin ein geeignetes Test und Visualisierungsverfahren auszuwählen um die korrekte Funktion des Programms zu zeigen. Am Ende haben wir uns für eine Kombination aus visueller Darstellung (grafische Ausgabe des gefilterten sowie des ungefilterten Signals) und Überprüfung durch Nachrechnen (mittels eines C-Programms, welches die Werte auf gleiche Weise sortiert und anschließend vergleicht) entschieden, da so die korrekte Funktion am besten nachvollzogen werden kann.

Natürlich kam es auch zu einigen kleineren Entscheidungen, wie zum Beispiel der Wahl des Layouts für die Spezifikation, oder Zielsetzungen für das nächste Treffen. Diese wurden jedoch ebenso wie die beiden eben genannten großen Entscheidungen stets schnell und unkompliziert in der Gruppe gelöst.

Positive und negative Ereignisse:

Positive Ereignisse:

Insgesamt verlief das Projekt überaus reibungslos. Es wurden immer alle Termine für Treffen eingehalten, die Kommunikation außerhalb der Treffen war mittels WhatsApp schnell und unkompliziert, und sobald ein Gruppenmitglied auf ein Problem gestoßen ist, gab es meist schnell eine gute Antwort von einem der anderen Gruppenmitglieder.

Nicht nur die Termine für Treffen, sondern auch die Vereinbarten Meilensteine wurden stets zuverlässig erreicht, sodass unsere Gruppe eigentlich nie im Zeitverzug war. Des Weiteren gab es eine überaus gute Aufgabenverteilung. Kannten sich zum Beispiel ein oder zwei Gruppenmitglieder nicht so gut mit der Programmierung in C oder mit LaTeX aus, so hat ein anderes Gruppenmitglied diese Aufgabe stets zuverlässig übernommen. Genaueres zur Aufgabenverteilung finden sie im gleichnamigen Unterkapitel.

Negative Ereignisse / Probleme:

Während des Projekts kam es eigentlich zu keinen wirklichen negativen Ereignissen. Eine der wenigen Sachen welche jedoch besser hätte laufen können, war zu Beginn der Implementierung. Wie bereits in ‚Entstehung der Lösung‘ beschrieben, haben wir hier zu Beginn noch an zwei unterschiedlichen Versionen gleichzeitig gearbeitet, von denen wir am Ende jedoch nur eine Weiterverfolgt haben. Hier hätten wir etwas Zeit und Arbeit gespart, wenn wir direkt nur an einer Version gearbeitet hätten. Da wir jedoch bei der Implementierung frühzeitig fertig wurden, und in keinerlei Zeitschwierigkeiten kamen, ist dieses negative Ereignis als nicht wirklich gravierend einzustufen.

Probleme gab es während des Projektverlaufs auch vergleichsweise wenige. Trat eines auf, so wurde es stets zuerst über WhatsApp angesprochen. Konnte hier keine schnelle Lösung gefunden werden, so wurde es im nächsten Gruppentreffen besprochen. Konnte das Problem dann immer noch nicht geklärt werden, so wurde ein Treffen mit dem Tutor beschlossen. Dies kam während des Projekts allerdings nur ein einziges Mal, bei der Frage der Verwendung von 32 Bit Registern vor. Dieses Treffen wurde dann auch dazu genutzt, sich über kleinere Aspekte, welche vorher bereits in der Gruppe besprochen wurden, mit dem Tutor rück zu versichern. Sonstige kleinere Probleme welche während des Projekts auftraten umfassten die Wahl eines Layouts für die Spezifikation, die Entscheidung für einen Sortieralgorithmus, die Strukturierung gewisser Untermethoden, oder die Auswahl einer Testmethode. Diese wurden jedoch stets schnell gruppenintern gelöst.