



Santa Claus Problem

NVS Projekt 1

Alexander Grill 5CH1F
12. Januar 2021

Informatik
HTBLUvA Wr.Neustadt
Österreich

Inhaltsverzeichnis

1	Einführung	2
1.1	Vorwort	2
1.2	Motivation	2
2	Aufgabenstellung	3
2.1	Erleuterung der Grundproblematik	3
2.2	Idee	4
2.3	Kommandozeilenparameter	4
2.4	Konsolen Ausgabe	4
3	Themenbereiche	5
4	Implementierung	6
4.1	Aufbau	6
4.2	Dokumentation des Source Codes	6
4.2.1	Klassen	6
4.2.2	Funktionen	6
4.2.3	Sonstiges	6
4.3	Verwendete Bibliotheken	6
4.3.1	CLI11	6
4.3.2	spdlog	6

1 Einführung

1.1 Vorwort

Auf Grund der aktuellen Lage(COVID-19) in Österreich, wurde der Unterricht an den Schulen in Form von Distance-Learning abgehalten. Deshalb war die Durchführung der Praktischen Arbeit in Netzwerktechnik nicht möglich. Herr Professor Kolouseck gab uns daraufhin eine Projektarbeit die bis zum 20.01.2021 zu erledigen ist. Die Gesamtnote des Projekt ersetzt ausschließlich die Note, die man bei der Praktischen Arbeit erworben hätte. In diesem Projekt geht es darum mit Prozessen, Thread und Synchronisation für die jeweiligen Anwendungsszenarien entwickelt werden, um damit zu zeigen, dass die Praktischen Fähigkeiten zum Implementieren verteilter Systeme erworben wurden. Die Programmiersprache die für die Implementierung verwendet wird, ist C++.

Das Projekt besteht aus 3 wichtigen Punkten:

- praktischen Ausarbeitung
 - dabei wird die Grundproblematik simuliert, klargestellt und verschiedene Szenarios angewendet
- theoretisch Ausarbeitung
 - in diesem Teil wird die Aufgabenstellung, Probleme, Dokumentation der Source Codes usw. festgehalten
- Git Hub Repository
 - das Projekt muss auf GitHub gehostet werden, um die Verwaltung der Projekts zu erleichtern und, um den Workflow zu dokumentieren
 - Commits müssen gemacht werden, das man etwas als Patch verwenden kann (Fehler, Probleme können zurückgenommen werden)
 - Commit-Meldungen sollen kurz und prägnant sein und sollen ausdrücken für was dieser Commit steht.

1.2 Motivation

In diesem Projekt wird die Santa Claus Problematik erläutert und mit Hilfe eines C++ Projekt simuliert. Santa Claus wird wegen zwei Faktoren geweckt, sonst benötigt er unbedingt seinen Schlaf. Er wird geweckt, wenn Elfen ihm brauchen, weil sie mit der Arbeit nicht weiter kommen, und dadurch die Produktion der Geschenke für die Kinder verlangsamt oder gar gestoppt wird. Deshalb ist es wichtig, dass in diesem Moment Santa geweckt wird, um Santa den Elfen helfen zu können. Zugleich kommen auch die Rentiere in unterschiedlichen Zeitpunkten zurück von ihrer Reise und sammeln sich im Stall. Dort warten sie gemeinsam bis sie vollzählig sind und Santa mit ihnen, folgedessen die Geschenke zu den Kindern liefern. Auch in diesem Moment ist es äußerst wichtig, dass Santa geweckt wird, weil er schon einen Teil der Geschenke mit den Rentieren ausliefern kann. Zu beachten ist jedoch, dass es zu keinem Zusammenstoß zwischen den Gruppen kommt und Santa Claus nicht weiß, welche Tätigkeiten zuerst vollendet werden sollen bzw. welche eine höhere Priorität haben.

2 Aufgabenstellung

2.1 Erleuterung der Grundproblematik

Dieses Problem stammt aus William Stallings Operating Systems. Dabei wird folgende Problemstellung beschrieben:

Santa Claus sitzt in seinem Spielwarenladen am Nordpol und schläft, während seine zurückgekehrten Rentiere im Stall fressen um Kräfte für die jährliche Auslieferung der Geschenke an die Kinder zu Weihnachten zu sammeln. Seine fleißigen Elfen arbeiten sorgfältig an den Geschenken der Kinder in der Spielzeugfabrik. Hin und wieder kann es vorkommen, dass die Elfen beim Basteln vor einem Problem stehen und ohne Hilfe vom Santa nicht mehr weiter machen können. Es wäre eine Katastrophe, wenn die Geschenke nicht rechtzeitig am Heiligen Abend an die Kinder ausgeliefert werden können, weil einige Elfen die Produktion der Geschenke blockiert haben.

Aus diesem Grund muss Santa Claus unbedingt geweckt werden, obwohl er ein gewisses Maß an dringendem Schlaf benötigt. Erst wenn 3 oder mehr Elfen ohne Hilfe nicht mehr weiter arbeiten können wecken sie ihn auf. Wenn drei Elfen ihr Problem gelöst haben, müssen alle anderen Elfen, die den Weihnachtsmann besuchen wollen auf die Rückkehr dieser Elfen warten. Jedoch ist zu beachten, dass wenn Elfen vor der Tür seines Ladens warten, während das letzte Rentier für die Auslieferung aus dem Tropen zurückkehrt, beschließt der Weihnachtsmann, die Elfen bis nach Weihnachten warten zu lassen. Denn die es ist wichtiger den Schlitten fertigzustellen und die Pakete auszuliefern. Nachdem Santa Claus ihnen beim Problem geholfen hat, können die Elfen erleichternd weiterbasteln und Santa Claus seinen Schlaf forsetzen.

Allerdings möchte Santa Claus die Kinder nicht zu lange auf ihre Geschenke warten lassen, und deshalb soll er auch aufgeweckt werden, wenn genug Rentiere bereit sind bzw. zurück aus ihrem Urlaub im Südpazifik gekommen sind, um den Schlitten mit den Geschenken zu ziehen und eine Ladung Geschenke zu verteilen. Das letzte Rentier, das ankommt, muss den Weihnachtsmann holen, während die anderen Rentiere gemütlich in der warmen Hütte warten, bevor sie vor dem Schlitten gespannt werden.

Die restliche Zeit kann Santa schlafen, um für die nächste anstrengenden Tätigkeiten Kräfte zu schöpfen. Man kann davon ausgehen, dass stets genug Arbeit für die Elfen und genug Geschenke für eine Ladung Geschenke vorhanden sind. Das heißt Elfen und Rentiere sind weitgehend unabhängig voneinander. Santa Claus, seine Rentiere und die Elfen müssen jeweils durch einen eigenen Thread umgesetzt werden, sodass das unterschiedlichen Eintreten von Szenarien festgehalten werden kann.

2.2 Idee

Das Programm besteht aus drei Threads nämlich: SantaClaus, Rentiere, Elven. Der Thread SantaClaus schläft so lange bis, entweder alle seine benötigten Rentier zurückgekommen sind, oder 3 oder mehrere Elven ihm drigend brauchen. Im Thread Rentier wird modiliert, dass noch einer zufälligen Dauer, Rentiere zurückkommen und Santa danach aufwecken, sodass sie schon einen Teil der Geschenke an die Kinder liefern können. Im Thread Elven wird ebenso modiliert, dass nach einer zufälligen Zeit Elven Santa um seine Hilfe bitten. Nach einer bestimmten Anzahl von Elfen wird auch hier Santa geweckt, sodass sie weiter arbeiten können. Das wichtige ist vorallem, dass die Geschenke bis zum Heiligen Abend ausgeliefert werden können und, dass wenn alle Rentier da sind, Santa auf die Hilferufe der Elfen verzichtet und die Geschenke ausliefert.

2.3 Kommandozeilenparameter

Mit Hilfe von Kommandozeilenparameter soll dem Benutzer ermöglicht, werden die Anzahl der Rentier und der Elfen zu definieren. Diese zwei Zahlen legen die maximal benötigte Anzahl fest, um den Thread Santa Claus aus seinem Schlaf zu holen. Ebenso kann der Benutzer auch die Zeit in Stunden bis zu Weihnachten angeben, dadurch kann geprüft werden ob es sich zu geringer Zeit ausgeht die Pakete bis zu Weihnachten auszuliefern. Es werden keine negativen Zahlen, Buchstaben, Sonderzeichen akzeptiert und wenn der Benutzer sonstige Hilfe braucht kann er sich die Information mit dem Kommandozeilenparameter holen.

-r,- -r ... number of reindeer, which will be needed to fly

-e,- -e ... number of elves, which will be needed to wake up Santa

-t,- -t ... number of hours until christmas

-tab,- -tab ... number of hours until christmas

-h,- -h ... help

2.4 Konsolen Ausgabe

Die Abarbeitung bzw. Resultate der Threads SantaClaus, Elven und Rentiere werden in der Kommandozeile ausgegeben. Somit kann der Benutzer gut nachvollziehen, welche Tätigkeiten das Programm abgewickelt hat, welche noch bevor stehen und ob es zu Fehler-situation oder Ausnahmen gekommen ist. Angenommen es kommen Rentier zurück oder Elven benötigen Santa's Hilfe so wird auch in der Kommandozeile dies wird mit spdlog festgehalten. Um am Ende einen genauen und übersichtlichen Überblick den Benutzer zu verschaffen, wird mit dem Kommandozeilenparameter -tab eine Tabelle ausgegeben die Zeigt, wie viele Rentiere im Stall sind, wie viele Elfen Hilfe brauchten und wie viele Stunden Santa Claus geschlafen hat.

3 Themenbereiche

Das Projekt umfasst folgende Themengebiete, die unter anderem in folgenden Folien gut dokumentiert sind:

- 10_processes
- 11_threads
- 12_threads2
- 13_synchronization
- 14_condition_synchronization
- 15_sync_mechanisms
- 16_threadsafe_interfaces
- 17_dist_sync
- 18_task_based_programming
- 19_parallel_programming
- 20_threads_perfmem
- 21_encoding
- 22_data_interoperability
- 23_character_encoding

4 Implementierung

4.1 Aufbau

4.2 Dokumentation des Source Codes

4.2.1 Klassen

4.2.2 Funktionen

4.2.3 Sonstiges

4.3 Verwendete Bibliotheken

4.3.1 CLI11

4.3.2 spdlog