

Die Chance einer Variablen

Es war ein Moment wie jeder andere. Als Gustav von dem üblichen Stromstoß mit den ersten Werten gefüllt wurde. Er ahnte noch nicht, was für ein Programmlauf es werden würde. Gustav ist eine Variabel in einem Buchungsprogramm.

Er ist nichts besonderes geworden. Sein Traum einmal eine wichtige globale Variabel zu werden, zerplatzte bei dem Einsatz in einer kleinen Unterfunktion, die die Aufsummierung einiger Kostenpunkte übernahm. Immerhin war er zuständig für das Sammeln der einzelnen Positionen und die Rückgabe an das aufrufende Programm. Damit war er schon wichtiger als die normalen Schleifenzähler, die oftmals Identitätskrisen erleiden, da es sie in zu vielen Modulen mit gleichen Namen gibt. Gustav konnte wenigstens von sich behaupten einmalig im System zu sein. Seine Arbeit verrichtete er zu aller Zufriedenheit. Hin und wieder waren seine Werte falsch gerundet, fielen aber wegen einer nachträglichen Formatierung nicht weiter auf. Für seinen Posten bekam er einen 32 Bit Adressraum zugewiesen, die er geschmacklich mit einer Mantisse eingerichtet hat. Durch seine zwei Nachkommastellen konnte er das immer rechtfertigen. Gustav konnte sich nicht beschweren als Double - Variabel genießt man mehr Ansehen als ein Integer, da Fließkommaberechnungen immer mehr Verantwortung bedeuten als Ganzzahlarithmetik. Als Short oder Integer - Variable schafft man mit Mühe den Aufstieg zu einer Unsiegend aber nur mit Glück erhält man die Möglichkeit Nachkommastellen zu verwalten. Gustav hatte das Glück bei der Implementierung und wußte es auch zu schätzen aber dennoch hätte er gerne den Sprung zu einem Long Double mit einem verdoppelten Adressraum gemacht, aber das ist wohl nur ein Traum.

Gustav ist bereit die Kummulation auszuführen. Seine Kollegen sind bereit ihre Werte zur Verfügung zu stellen, damit er sie zu einem 5stelligen Gesamtwert mit zwei Nachkommastellen addieren kann. Gustav weiß gar nicht mehr wie oft er diese Aufgabe schon erfüllt hat. Er könnte es jederzeit erfahren wenn er den globalen Zähler mal auslesen würde, aber der ist wegen seiner stumpfsinnigen Arbeit meist recht mürrisch und deshalb nicht gerade das angenehmste Erlebnis, was eine Variabel haben kann. Für Gustav ist die Erfolgsquote seiner Arbeit auch nicht so wichtig.

Nach Programmlauf ist Gustav erst sehr spät an der Reihe, daher wird er oft vom Betriebssystem durch den Speicher geschickt und manchmal auf Festplatte ausgelagert. Das ist eine willkommene Abwechslung für eine Variable, da sie sonst immer nur den gleichen Adressraum zu füllen bekommt. Diese Abwechslung ist den globalen Variablen meist nicht vergönnt. Durch die ständige Gültigkeit und zwanghafte Zugriffsmöglichkeit werden globale Variablen ständig unter Streß gesetzt. Teilweise ist Gustav froh, diesen Freiraum zu haben.

Bei diesem Programmlauf wird Gustav vom Betriebssystem in einem stark benutzen Adressraum zwischen &0FFD und &0FFE abgeschoben. Hier treiben sich Literale und rücksichtslose Strings herum. Literale sind wegen ihrer Aggressivität fest im Programmcode eingesperrt. Das macht sie noch kontrollierbar, aber Strings kennen keine Grenzen. Ständig versuchen sie sich zu vergrößern, dabei ist es ihnen völlig gleich, welchen Schaden sie bei den anderen Variablen anrichten. Ohne Skrupel überschreiben sie Werte und riskieren einen Totalabsturz. In solchen Gebieten sollte man sich als gewissenhafte Variabel nicht aufhalten, sonst kann es zu gefährlichen Wertbelegungen kommen.

"Hast'e mal n' Bit ?" erklingt es immer in Gustavs Leitungen. Die Habgierigkeit der Strings kennt mal wieder keine Grenzen. Für die Ausdehnung im Adressraum nehmen die Strings auch in Kauf, daß brave Short - und gewissenhafte Double - Variablen ihre Werte verlieren. Die Konsequenzen müssen sie nicht tragen. Diese Gefrässigkeit der Strings macht sie unter allen Variablen sehr unbeliebt. Wie jede Variable glaubt auch Gustav an seinen Programmierer, der ihn geschaffen hat und ihm seine Aufgabe gegeben hat. In diesen Momenten der panischen Angst um seine Bits betet er zu ihm und hofft schnell in andere Adressbereiche zu kommen.

Gustav ist nochmal davongekommen. Das Betriebssystem schaufelte einen riesigen nicht benutzten Programmteil in den virtuellen Speicher auf der Festplatte. Es ist ein befreiendes Gefühl eine Pause von den ständigen Refresh-Stromstößen zu bekommen und stattdessen in einem magnetischen Feld zu versinken. Sehr oft kann Gustav dieses Gefühl nicht erleben, da das Programm nicht sonderlich groß ist und irgendwie immer in den Hauptspeicher paßt. So erfreut sich Gustav über die große Auslastung des Gesamtsystems, um in den Genuß des Magnetbades zu kommen.

Doch dann passierte etwas Unvorhergesehenes. Die Datenbank ist in einem inkonsistenten Zustand und fällt für die Belegung einer äußerst wichtigen globalen Variabel aus. Ein Notfallprogramm wird angestossen, in der Gustav die Vorbelegung dieser globalen Variablen übernehmen soll. Das war seine Chance eine wichtige Aufgabe zu übernehmen. Diese Nachricht sprach sich im Speicher herum wie eine Überspannung. Es gab viele Funktionen, die diese Aufgabe erfüllen konnte, aber die Wahl fiel auf Gustav. Neid und Mißgunst schlug ihm entgegen. Andere lokale Variablen, die ähnliche Rückgabewerte lieferten versuchten das Hauptprogramm in der Entscheidung für Gustav zu beeinflussen. Aber die if-Anweisung lieferte TRUE für Gustav. Gustav war ganz aufgeregt. Er hatte schon Angst, er würde versagen und das System zum Absturz bringen. Eiligst überprüft er seine untergebenen lokalen Variablen. Das Team stand bereit für die große Aufgabe. Die if-Anweisungen funktionierten und wählten die Variablen aus, die aufaddiert werden sollten. Gustavs dreißigstes Bit pochte, das letzte Bit fiel teilweise ganz aus. Hoffentlich wird der hohe Adressbereich nicht benötigt. Jeden Moment sollte der Aufruf call by value erfolgen. Drei Parameter gilt es zu verarbeiten und einen Wert zurückzuliefern. Die Spannung war nicht zu ertragen. Die Schleifenvariable, die ohnehin nicht das beste Nervenkostüm hat, initialisierte sich gleich drei mal. Der formale Short - Parameter, der die Werte aufnehmen soll, konnte ein ganzes Byte nicht unter Kontrolle bringen. Wenn das bei der Übergabe passiert. Jetzt heißt es zusammenreißen und tief durchleiten.

Plötzlich ist der Aufruf da. Der formale Parameter kann die Variable noch rechtzeitig einer Hilfsvariabel übergeben, bevor er völlig zerlöscht ist. Sieben, Drei Komma Fünf, Neun. Gustav schluckt. Der Angststrom steht ihm bis zum 23.Bit. Was ist das für eine Kombination ? Hatte er diesen Aufruf jemals gehabt ? Die lokalen Variablen schauten sich fragend an. Nach und nach stellten sie fest, daß nichts zu addieren war. Jeder einzelne wurde von den if-Anweisungen ignoriert. Gustav bekam Zweifel, sein Anfangswert war entscheidend. Auf ihn kam es an. Bis jetzt übernahm er immer die Anfangswerte seiner Untergebenen. War er initialisiert ? Sollte die große Aufgabe daran scheitern, daß er seinen Ursprung nicht kennt ? Gab ihm sein Programmierer ein lückenlosen Werteverlauf mit ? Gustav versucht sich krampfhaft an den Wert zu erinnern, der seine Existenz und die seiner Kollegen vor dem blauen Tod im Sumpf des Betriebssystems bewahren kann.

Neid und Mißgunst sind vergessen. Alle sitzen im gleichen Programm. Keiner seiner Kollegen in den anderen Funktionen kann wirklich sicher sein immer definiert zu sein. Die Panik verbreitet sich im Speicher. Alle globalen und lokalen Variablen egal ob Strings, Doubles oder Shorts versuchten Gustav zu helfen bei der Suche nach seiner Initialisierung. Aber Gustav brauchte ihre Vorbelegungen nicht, er lieferte 0.0.

Das Programm hatte den Notfall erfolgreich überstanden. Gustav wurde wegen guter Arbeit mehr Speicherplatz zugewiesen und war somit Long Double. Keiner seiner Kollegen mißgönnte ihm diesen Erfolg, da viele erkannt haben, daß sie nicht initialisiert waren. Und wenn das Programm noch nicht abstürzt ist so rechnet es noch heute.

Nov. 2002 , Volker Stang