

Anhang: Elementare Eigenschaften der Computerhardware

Speicher und Adressräume

Heute muss nur noch in ganz seltenen Fällen maschinennah programmiert werden. Über 99 % aller Programmtexte werden heute in höheren Programmiersprachen formuliert. Dennoch sollten auch die Programmierer, die nur in höheren Sprachen programmieren, noch eine grobe Vorstellung davon haben, wie es in der Hardware des Computers aussieht. Denn alles, was aufgrund der Programmabwicklung geschieht, ist Veränderung von Binärwörtern an bestimmten Orten im Computer, wobei diese Orte meistens Speicherzellen und manchmal Kanäle sind. Im Bild sind die unterschiedlichen Orte für Binärwörter zusammengestellt, die man in einem Prozessor unterscheiden kann.

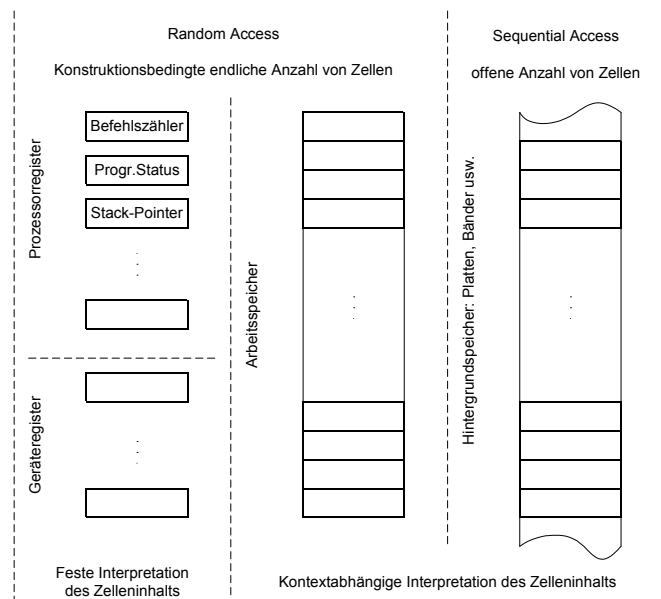


Bild Unterscheidung der Orte für Binärwörter in der Computerhardware

Die Orte sind nach zwei Kriterien klassifiziert. Das eine Kriterium beruht auf der Frage, ob die Interpretationsvorschrift für ein Binärwort an einem solchen Ort kontextunabhängig festliegt oder ob diese Interpretation kontextabhängig ist. Das andere Kriterium bezieht sich auf die Art des Zugangs zu dem betrachteten Ort. Von Random Access, was im Deutschen wahlfreier Zugriff heißt, sprechen wir, wenn der Aufwand für den Zugang zu dem Ort unabhängig davon ist, welcher Ort unmittelbar vorher angesprochen wurde. Zur Veranschaulichung des wahlfreien Zugangs denke man an den Vortragenden im Hörsaal und seine Studenten, die er vor sich sitzen hat. Er kann auf jeden einzelnen Studenten zeigen, und der Aufwand, den er für das Zeigen auf einen Studenten erbringen muss, ist unabhängig davon, auf welchen Studenten er zuvor gezeigt hat. Anders liegt der Fall bei Orten, für die es nur einen sequentiellen Zugang gibt. Denken Sie an einen sehr langen Gang, von dem auf der einen Seite Türen abgehen, hinter denen jeweils ein potentieller Gesprächspartner sitzt. Den Aufwand, den ich treiben muss, um zu einem bestimmten Gesprächspartner zu gelangen, ist abhängig davon,

mit welchem Gesprächspartner ich zuletzt Kontakt hatte. Am einfachsten ist es, wenn ich die Gesprächspartner nacheinander der Reihenfolge nach besuche. In der Technik handelt es sich bei Speichern mit sequentielltem Zugang um Bänder oder Platten, die an einem Schreib/Lesekopf vorbeibewegt werden müssen.

In Bild 89 ist gezeigt, dass alle Zellen im sequentiellen Zugriff (Sequentiell Access Mode SAM) kontextabhängig interpretiert werden. Die Kontextabhängigkeit der Interpretation bedeutet, dass man einem Binärwort aufgrund seines aktuellen Ortes nicht eindeutig eine Bedeutung zuordnen kann, sondern dass die Bedeutung eines Binärwortes an einem bestimmten Ort noch davon abhängt, welche Binärwörter an bestimmten anderen Orten zur gleichen Zeit vorliegen.

In einem Computersystem gibt es nur sehr wenige Orte mit kontextunabhängiger Interpretation. Zum einen sind dies die sogenannten Prozessorregister, und zum anderen sind dies die Gerätereister. Obwohl man in beiden Fällen von Registern spricht, handelt es sich nicht in beiden Fällen um reguläre Speicherzellen. Die Gerätereister sind zum Teil gar keine Speicherzellen, sondern Kanäle. Alle Orte im Bild außer den Gerätereistern sind normale Speicherzellen. Eine solche Speicherzelle verhält sich wie eine Tafel, auf die man etwas schreiben kann, was man später wieder lesen kann. Die Gerätereister dagegen sind die Kanäle, die den Computer mit den peripheren Geräten verbinden. Jedes Bit in einem Gerätereister hat eine gerätespezifische Bedeutung. So könnte es beispielsweise ein Bit geben zum Anschalten des Lichts im Hörsaal. Wenn der Programmierer will, dass das Licht angeht, wird er eine 1 an diese Binärstelle schreiben. Ein anderes Gerätereister könnte dazu dienen, dem Computer die aktuelle Uhrzeit anzuzeigen. In diesem Falle ändert sich das Binärwort an diesem Ort, ohne dass per Programmabwicklung etwas geschrieben wird.

Das bekannteste Prozessorregister ist der Befehlszähler. Das darin enthaltene Binärwort wird stets als Adresse einer Zelle im Arbeitsspeicher interpretiert, und der Inhalt der so adressierten Arbeitsspeicherzelle wird als Maschinenbefehl interpretiert. Wenn ein Programm abzuwickeln ist, welches keine Sprunganweisungen enthält, wird der Befehlszähler jeweils um 1 erhöht, wenn der nächste Befehl geholt werden soll. Bei Sprungbefehlen steht in der Sprunganweisung eine neue Zieladresse, die in den Befehlszähler zu übertragen ist. Ein weiteres Prozessorregister mit sehr einfacher Interpretation ist der Stackpointer. In jedem Prozessor muss es mindestens einen Hardwarestack geben, für den es ein Register gibt, welches in jedem Falle als Stackpointer interpretiert wird. Zusätzlich kann sich auch noch jeder Programmierer eigene Stacks anlegen. Im Unterschied zum Hardwarestack liegen bei den programmierten Stacks auch die Stackpointer an Orten im Arbeitsspeicher und sind nur durch kontextabhängige Interpretation als Stackpointer zu identifizieren.

Das Programmstatuswort (PSW)

Das Binärwort im Programm-Status-Register wird abschnittsweise interpretiert, d.h. in diesem Fall gibt es keine geschlossene Interpretation wie beim Befehlszähler oder beim Stackpointer. Obwohl die modernen Prozessoren noch mehr Abschnitte im Programmstatuswort haben, betrachten wir hier nur die drei ursprünglichen:

Verzweigungsbits / Unterbrechbarkeitsniveau / Privilegienstatus

Zur Realisierung von Fallunterscheidungen und Wiederholungen muss man die Möglichkeit haben, Bedingungen abzufragen. In den höheren Programmiersprachen kann man diese Be-

dingungen frei formulieren, d.h. dort gibt es kein festes vorgegebenes Bedingungsrepertoire. Bei der Konstruktion des Prozessors aber muss sich der Konstrukteur auf eine endliche Menge von Bedingungen festlegen, denn das was später die Programmierer als Bedingung in ihren Programmen brauchen, ist eine offene Bedingungswelt, die es so in der Hardware nicht geben kann. Im Programmstatuswort gibt es verhältnismäßig wenige Binärstellen, die in den bedingten Sprunganweisungen abgefragt werden können. Jede Ausführung einer Anweisung, die ein Ergebnis liefert, führt zu einer Belegung von Bedingungsbits. Nicht jede Befehlsausführung liefert ein Ergebnis, insbesondere die Sprungbefehle gehören zu den nichtergebnisliefernden Befehlen. Dagegen gehört die Kopieranweisung, die verlangt, dass eine Kopie des Binärwortes aus der Zelle a in die Zelle b geschrieben wird, zu den ergebnisliefernden Anweisungen. In den Bedingungsbits gibt es das sogenannte Zerobit, welches immer dann auf 1 gesetzt wird, wenn sich bei einer Befehlsausführung ein Nullwort ergibt. Ein anderes Bit im Programmstatuswort ist das Negativbit; dieses wird auf 1 gesetzt, wenn sich bei einer Befehlsausführung eine negative Zahl ergibt. Bei der Zweierkomplementcodierung äußert sich eine negative Zahl immer darin, dass ihr links äußerstes Bit auf 1 steht.

Die manchmal recht komplizierten Prädikatsausdrücke in den Fallunterscheidungen und Wiederholungen in den höheren Programmiersprachen müssen vom Übersetzer in Programmstücke übersetzt werden, an deren Ende ein bedingter Sprung steht, der sich auf ein ganz bestimmtes Bedingungsbit im Programmstatuswort bezieht. Dieses muss im voranstehenden Programmstück so gewonnen worden sein, dass es der komplizierten Abfrage entspricht. Will man beispielsweise abfragen, ob in der Zelle i eine Zahl steht, die größer ist als 17, wird man eine Subtraktion durchführen, die das Ergebnis $(i-18)$ liefert. Wenn dieses Ergebnis negativ ist, ist i selbst nicht größer als 17.

Das Unterbrechbarkeitsniveau

Als nächsten Abschnitt des Binärwortes im Programmstatuswort, der eine eigene Interpretation hat, betrachten wir das sogenannte Unterbrechbarkeitsniveau. Seit über 30 Jahren werden die Prozessoren so gebaut, dass eine laufende Programmabwicklung unterbrochen werden kann zugunsten des Beginns der Abwicklung eines anderen Programms. In der Anfangszeit der Computerei gab es die Unterbrechbarkeit noch nicht. Aber auch damals gab es schon die Ereignisse, die eine weitere Abwicklung des laufenden Programms unmöglich machten. Als einfachstes Beispiel denke man an den Versuch einer Division durch Null. Wenn früher solch ein Ereignis vorkam, blieb der Rechner einfach stehen und zeigte den Abbruchgrund auf dem Bedienfeld an. Selbstverständlich blieb der Rechner auch stehen, wenn er ans reguläre Programmende gekommen war. Das reguläre Programmende war im Arbeitsspeicher durch einen Stopbefehl gekennzeichnet. Heutzutage bleibt ein Rechner nur noch stehen, wenn man ihn ganz bewusst abschaltet. Anderenfalls läuft er dauernd durch, d.h. er führt dauernd Befehle aus, auch dann, wenn er eigentlich nichts zu tun hat. Dies wurde dadurch möglich, dass die Hardware der heutigen Prozessoren die Unterbrechbarkeit der Programmabwicklung realisiert.

Man muss zwischen zwei unterschiedlichen Unterbrechungsgründen unterscheiden, den synchronen und den asynchronen Unterbrechungsgründen. Synchroner Unterbrechungsgründe werden auch als Ausnahmesituationen oder Exceptions bezeichnet. Sie sind dadurch gekennzeichnet, dass bei einem Versuch, eine Anweisung auszuführen, eine Ausnahmesituation festgestellt wird, die das reguläre Ausführen der Anweisung unmöglich macht. Solche Ausnahmesituationen gibt es auch im Alltag. Man denke an den Fall, dass man mit einem Kugel-

schreiber schreiben will und während des Schreibens feststellt, dass die Patrone leer ist. In diesem Falle muss man die beabsichtigte Tätigkeit zwangsläufig abbrechen und sich einer anderen Aufgabe zuwenden. Asynchrone Unterbrechungsereignisse heißen asynchron, weil sie nichts mit der aktuellen Befehlsabwicklung zu tun haben. Es handelt sich um Meldungen aus der Peripherie des Rechners. Typische asynchrone Meldungen sind beispielsweise die Meldungen des Zeitgebers, die für den Rechner das gleiche bedeuten wie für uns das Klingeln eines Weckers.

Im Programmstatuswort wird in Form des sogenannten Unterbrechbarkeitsniveaus jeweils festgehalten, welche asynchronen Unterbrechungsgründe aktuell zu einer Unterbrechung führen können und welche nicht. So könnte ich beispielsweise zu meiner Sekretärin gesagt haben: Ich habe jetzt eine wichtige Besprechung und möchte nicht gestört werden. In diesem Falle wird sie nur noch ganz wenige Unterbrechungsgründe akzeptieren, nämlich wenn beispielsweise der Bundespräsident mich sprechen will oder wenn sie mir mitteilen muss, dass das Gebäude brennt. In Bild 90 sind die Verhältnisse veranschaulicht.

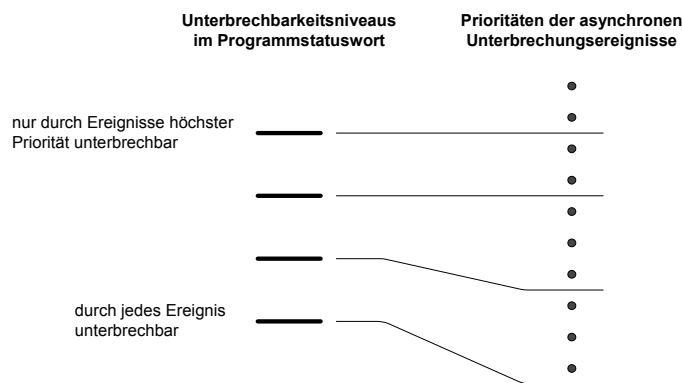


Bild 90 Veranschaulichung der Unterbrechbarkeitsniveaus

Links sehen wir vier Unterbrechbarkeitsniveaus und rechts sehen wir die nach Prioritäten geordneten Unterbrechungsereignisarten. Wenn das niederste Unterbrechbarkeitsniveau im Prozessorstatuswort eingetragen ist, bedeutet dies, dass jedes asynchrone Unterbrechungsereignis zu einer Unterbrechung führt. Wenn dagegen das höchste Unterbrechbarkeitsniveau im Programmstatuswort eingestellt ist, kann der Rechner nur noch durch ganz extreme Ereignisse unterbrochen werden. Ein solches extremes Ereignis ist die Feststellung, dass eine Spannungsunterbrechung vorliegt. In diesem Fall müssen die verbleibenden Millisekunden, die der Rechner noch aus den in den elektromagnetischen Feldern gespeicherten Energien versorgt wird, dazu genutzt werden, einen Zustand auf dem Festspeicher herzustellen, der eine problemlose Fortführung der unterbrochenen Arbeit ermöglicht, nachdem die Stromversorgung wieder funktioniert.

Vor der Einführung der Unterbrechungstechnik gab es keine Betriebssysteme, denn Betriebssysteme dienen dazu, den Speicher und die Prozessorleistung auf unterschiedliche Programme zu verteilen, und dies setzt die Unterbrechungstechnik voraus. Die Unterbrechungstechnik hat auch das sogenannte Timesharing möglich gemacht, bei dem etliche unterschiedliche Programme sogenannte Zeitscheiben auf dem Rechner bekommen, so dass der Eindruck entsteht, alle diese Programme liefen parallel.

Der Privilegienstatus

Als dritten Abschnitt im Programmstatuswort betrachten wir den sogenannten Privilegienstatus. Die Möglichkeit, den Prozessor und den Speicher zur Abwicklung unterschiedlicher Programme pseudonebenläufig nutzen zu können, brachte die Notwendigkeit mit sich, die verschiedenen Programmabwicklungen gegeneinander abzuschotten. Das geschieht dadurch, dass viele Aktivitäten, die ein Programm verlangt, nicht mehr von Befehlen des Programms unmittelbar ausgelöst werden können, sondern nur noch über den Umweg des zwischengeschalteten Betriebssystems. So musste man insbesondere den einzelnen Anwendungsprogrammen den direkten Zugang zu den Geräteregeleinheiten verwehren. Es würde sich ein heilloses Durcheinander ergeben, wenn jeder Programmierer die Möglichkeit hätte, in seinem Programm auf die Geräteregeleinheiten zu schreiben. Man muss ja bedenken, dass im Abstand von wenigen Millisekunden sehr unterschiedliche Programmstücke zur Abwicklung kommen. Da kann man nicht zulassen, dass jedes dieser Programme direkt mit dem Drucker kommuniziert, denn sonst würden sich auf der Druckerausgabe unterschiedlichste Inhalte überlagern. Die Möglichkeit, den Programmen den Zugang zu bestimmten Orten zu verwehren, realisiert man dadurch, dass man den gesamten Befehlssatz des Prozessors in Teilmengen ordnet, wie dies im Bild gezeigt ist. Nur die Befehle in der innersten Teilmenge stehen dem normalen Benutzer für seine Programme zur Verfügung. Die zusätzlichen Bereiche enthalten Befehle, die nur noch ein Systemprogrammierer diszipliniert benutzen darf. Dass man hier noch weitere Klassen unterscheidet, wird hier nicht näher begründet.

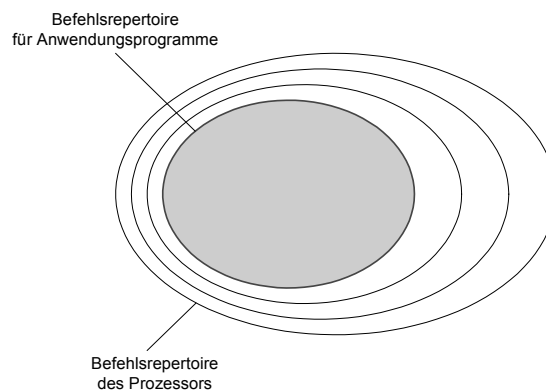


Bild Veranschaulichung des Privilegienstatus zur Unterteilung des Befehlssatzes

Bezüglich der Speicherverwaltung wurden die Verhältnisse gegenüber den Anfangsjahren der Computerei auch immer komplizierter. Anfangs befasste sich der ganze Rechner jeweils nur mit einem einzigen Programm, und der Programmierer hatte den gesamten Arbeitsspeicher in seinem Zugriff. Falls er dynamische Datenstrukturen realisieren wollte, musste er die Speicherverwaltung selbst programmieren. Später kamen Programmiersprachen hinzu, die dem Anwendungsprogrammierer die Programmierung der Speicherverwaltung abnahmen. Wenn man also mit NEW Speicherplatz für einen Record anfordert, muss man sich heute als Anwendungsprogrammierer nicht mehr darum kümmern, wo dieser Speicherplatz real im Arbeitsspeicher liegt (siehe hierzu das Bild 53).

Virtuelle Adressräume durch „Paging“

Damit nun auch noch die pseudonebenläufige Abwicklung unterschiedlichster Programme möglich wird, hat man noch eine zusätzliche Art von Speicherverwaltung eingeführt, die die Bezeichnung „virtuelle Adressierung“ erhielt. Die einzelnen Anwendungsprogrammierer inklusive der Compilerschreiber sollten sich nicht darum kümmern müssen, dass ja der reale Arbeitsspeicher für unterschiedliche Programme gleichzeitig gebraucht wird. Deshalb hat man den sogenannten virtuellen Adressraum eingeführt als zusammenhängenden Adressraum für den einzelnen Nutzer. Auch das Betriebssystem erhielt einen solchen virtuellen Adressraum, der vom Systemprogrammierer wie ein Arbeitsspeicher angesehen wird. Durch eine entsprechende Erweiterung der Hardware des Prozessors und durch entsprechende Prozeduren im Betriebssystem wird dafür gesorgt, dass diese unterschiedlichen virtuellen Adressräume auf den realen Arbeitsspeicher und den Hintergrundspeicher abgebildet werden. Es findet während der Abwicklung immer wieder ein Auslagern oder Einlagern von sogenannten Seiten (englisch *pages*) statt. Den beteiligten Hintergrundspeicher bezeichnet man auch als „Paging Device“. Eine Seite ist ein zusammenhängender Speicherbereich von 2^p kByte. In aktuellen Systemen ist p eine Zahl zwischen 0 und 3, d.h. in aktuellen Systemen liegen die Seitengrößen zwischen 1 und 8 kByte.