RECHENZENTRUM TH MÜNCHEN
ARBEITSGRUPPE BETRIEBSSYSTEME

INTERNSCHRIFT Nr. 57

THEMA Verwaltung benutzerspezifischer Daten auf Hintergrundspeichern

VERFASSER Plickert

DATUM 30.8.71

FORM DER ABFASSUNG

Entwurf
Ausarbeitung
Endform

SACHLICHE VERBINDLICHKEIT

Allgemeine Information
Diskussionsgrundlage
Erarbeiteter Vorschlag
Verbindliche Mitteilung
Veraltet

ÄNDERUNGSZUSTAND

BEZUG AUF INTERNSCHRIFTEN

42, 52

ANDERWEITIGE LITERATUR

Vorbemerkung

Die vorliegende Internschrift entstand als Beiprodukt bei der Vorbereitung auf den Vortrag "Verwaltung benutzerspezifischer Daten auf Hintergrundspeichern", der auf der 1. Jahrestagung der Gesellschaft für Informatik vom 12. bis 14. Oktober 1971 in München gehalten werden soll. Sie dient als Diskussionsgrundlage.

./. 2

Inhalt Seite

0.	Vortragskurzfassung	3
1.	Speicherverwaltung für benutzerspezifische Daten	5
1.1.	Voraussetzung und Problemstellung	5
1.2.	Virtueller Speicher als Lösungsansatz	5
1.3.	Speicherebenen	7
2.	Gebiete	10
2.1.	Dynamische Eigenschaften von Gebieten	11
2.2.	Adressierung von Gebieten	12
2.3.	Zugriff auf Gebiete	14
3.	Implementierung	15
3.1.	Logische Platte	15
3.2.	Behälter	16
3.3.	Systemdienste zur Verwaltung von Gebieten	17
3.4.	Adreßumsetzung für Gebiete	19
3.5.	Überlauf des Hintergrundspeichers	20
4.	Verfeinerung des Gebietskonzepts	21

./. 3

0. Vortragskurzfassung

Verwaltung benutzerspezifischer Daten auf Hintergrundspeichern

Es wird eine Speicherverwaltung für rotierende Hintergrundspeicher beschrieben. Ausgangspunkt sind die in Datenbasen und Dateien organisierten Informationsmengen, die auf dem Hintergrundspeichersystem (bzw. bei automatischer Verdrängung auf Magnetbändern) gelagert werden. Wenn man davon ausgeht, daß es keine apparative Unterstützung beim Satzzugriff (Satz = zugreifbare Einheit einer Datei) gibt und man uneingeschränkt Sätze in geordnete Satzbestände einschieben können will, ohne global reorganisieren zu müssen, wird vorgeschlagen, die Abbildung von Dateien auf den Hintergrundspeicher 2-stufig zu organisieren.
Auf der Zwischenschicht G der Transformationskette

Datei ⟶ G ⟶ Hintergrundspeicher

wird mit Hilfe der Organisationsform Gebiet das Einfügen, Streichen und Anhängen von Satzmengen buchungstechnisch gelöst.
Die Abbildung Gebiet ⟶ Hintergrundspeicher geschieht dadurch, daß ein Gebiet in Gebietsseiten, der Hintergrundspeicher in gleich großen Bereichen (Kacheln, page frames) organisiert ist, die einander zugeordnet werden.

Es wird die Adressierung, der Zugriff und das dynamische Verhalten von Gebieten beschrieben.

Das Überlaufproblem des Hintergrundspeichers wird durch eine automatische Verdrängung von Gebieten auf Magnetbänder gelöst. Verdrängungseinheit sind die in einem Behälter zusammengefaßten Gebiete eines Benutzers. Der Behälter erscheint unter einem anderen Aspekt als der Adreßraum für benutzerspezifische Daten.

./. 4

Die beschriebene Hintergrundspeicherverwaltung wird an der TR440 Installation des Leibniz-Rechenzentrums, München, implementiert.

./. 5

1. Speicherverwaltung für benutzerspezifische Daten

1.1. Voraussetzungen und Problemstellung

Wir setzen voraus, daß benutzerspezifische Daten in geordneten Satzmengen (= Dateien) organisiert sind. Diese Datenorganisationsform soll im wesentlichen "Karteicharakteristik" haben, d.h. Einschieben, Löschen und Anfügen von Sätzen soll ohne totale physikal. Reorganisation der Datei möglich sein, und bei Verarbeitung mit sequentiellem Zugriff (sequentielle Verarbeitung) einer mit random (Schreib)-Zugriff aufgebauten Datei soll die Zahl der Hintergrundtransporte ebenso groß sein, als wäre die Datei sequentiell aufgebaut. Ferner gehen wir davon aus, daß es keine besondere apparative Unterstützung beim wahlfreien Zugriff auf Sätze gibt und daß der Hintergrundspeicher, auf dem die Dateien liegen, blockorientiert ist (Block = n Wörter, kleinste physikalisch zugreifbare Speichereinheit).

Das Problem ist die Abbildung der Dateien auf den Hintergrundspeicher unter den Bedingungen:

- unbeschränktes Einschieben von Sätzen in die Datei ohne totale Umorganisation der Datei,

- dynamische Speicherzuteilung (d.h. Speicherzuteilung, die sich am aktuellen Bedarf orientiert)

- effiziente Speicherausnutzung (wenig Verschnitt).

1.2. Virtueller Speicher als Lösungsansatz

Die Forderung nach dynamischer Speicherzuteilung und die Tatsache der Verschiedenheit von Datenstrukturen und Speicherstrukturen führt natürlicherweise zum Be-

./. 6

griff des virtuellen Speichers, dessen Hauptaspekt die Unterscheidung zwischen Adreßraum und Speicherraum ist.

Als die beiden wichtigsten Methoden der Implementierung des virt. Speichers sind Segmentierung und Paging bekannt, die im Hinblick auf eine Hintergrundspeicherverwaltung folgendermaßen motiviert werden.

Die Segmentierung entsteht aus
- dem Wunsch nach Strukturierung der benutzerspezifischen Daten in z.B. Dateien,

- dem dynamischen Umfang von Dateien und

- Fragen der Mehrfachbenutzung von Dateien.

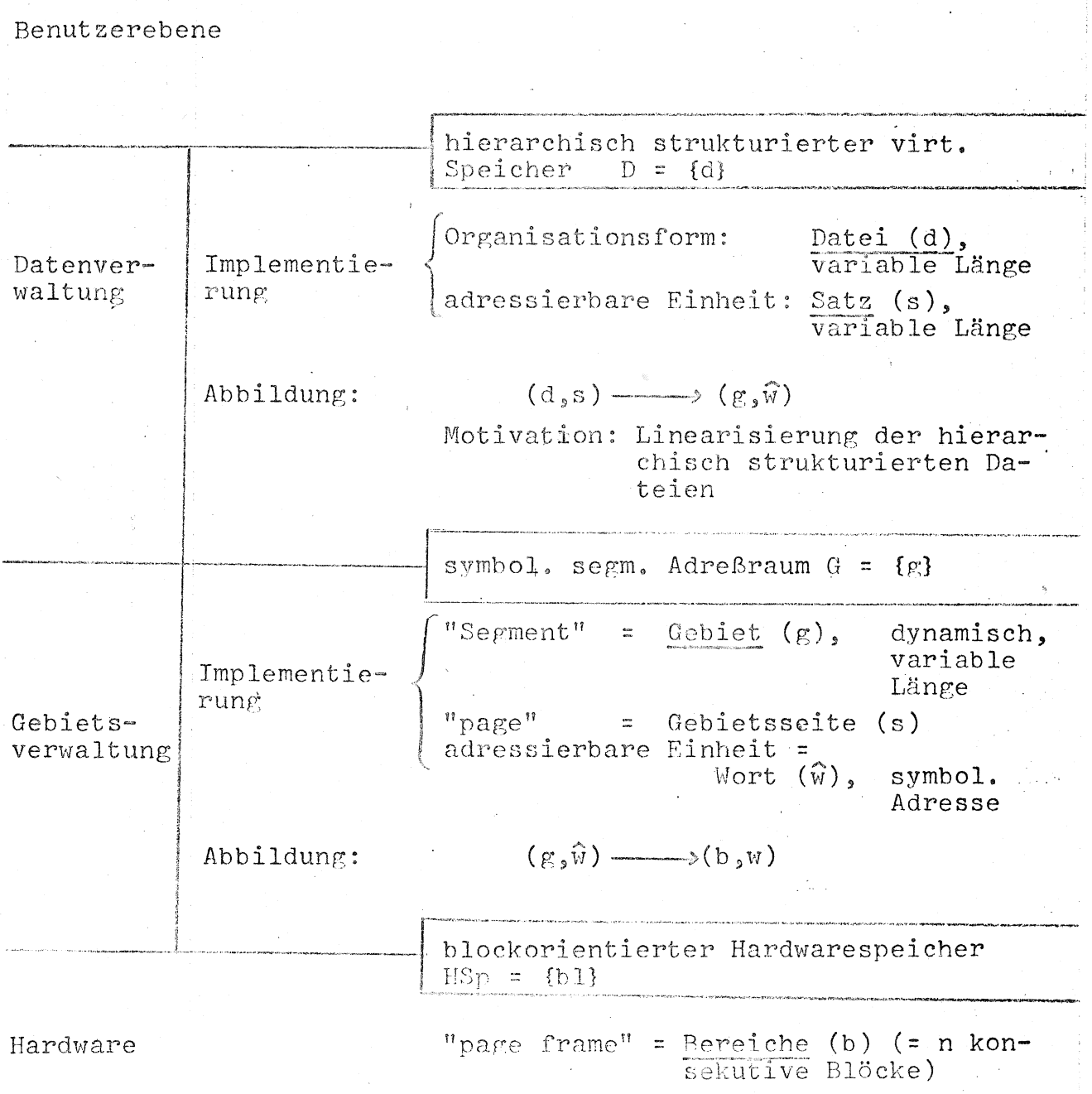
Während die Segmentierung eine inhaltsbezogene, logische Strukturierung des Adreßraums beinhaltet, steht die Paginierung (paging) unter dem Aspekt einer einfachen Speicherabbildung:

- Einteilung des Adreßraums und des physikal. Speichers in gleich große zusammenhängende Stücke, und Zuordnung dieser Stücke (pages ↔ page frames) zueinander.

Bei der Realisierung des virtuellen Hintergrundspeichers werden folgende Begriffe eingeführt.
Die Segmente des virtuellen Speichers werden Gebiete genannt. Gebietsseiten sind die "pages" und Bereiche sind die "page-frames" des Hintergrundspeichers.

Da es zweckmäßig ist, die Gebiete eines Benutzers in einer eigenen Organisationsform zusammenzufassen, führen wir den Begriff des Behälters ein:
Behälter sind benutzerspezif. virtuelle Hintergrundspeicher und Verdrängungseinheit bei Plattenüberlauf.

1.3. Speicherebenen (Text siehe nächste Seite)



./. 8

Das Strukturmodell, das dem Betriebssystem, in das die Verwaltung des Hintergrundspeichers eingebettet ist, zugrunde liegt, ist Dijkstras "Schichtenbild". Danach wird das laufende Gesamtsystem als eine Menge sequentieller Prozesse verstanden, die in Schichten eingeteilt sind. Da die Bedeutung eines Prozesses zweifach ist, nämlich die eines (Pseudo)prozessors und die eines Adreßraums, muß an jeder Schnittstelle zwischen zwei Schichten der Adreßraum, auf den die Prozesse der höherliegenden Schicht laufen, definiert werden.

Auf Benutzerebene legen die Benutzer ihre Daten in einen virtuellen Speicher, der gemäß den Datenstrukturen hierarchisch organisiert ist. Die rekursive Organisationsform Datenbasis und die nicht rekursive Organisationsform Datei, in der die benutzerspezifischen Daten in Form geordneter Satzmengen vorliegen, definieren eine Baumstruktur, mit deren Hilfe der Benutzer seine Datenmengen gliedern kann.

Die Transformation zur nächst tieferliegenden Schicht linearisiert die hierarchische Struktur. Die Schnittstelle definiert einen Adreßraum, der symbolisch segmentiert ist.
Die "Segmente" heißen Gebiete. Gebiete sind die zentrale Organisationsform bei der Abbildung von Dateien auf den physikalischen Speicher. Sie linearisieren die hierarchische Datenstruktur und realisieren die dynamische Speicheranforderung. Die Systemschicht, die Gebiete organisiert, heißt Gebietsverwaltung.

Gebiete selbst bestehen aus Gebietsseiten fester Länge, die geordnet sind. Die Speicheranforderung und Speicher-

belegung für Gebiete geschieht gebietsseitenweise, indem Gebietsseiten gleich großen zusammenhängenden Hardwarespeicherstücken, genannt Bereiche, zugeordnet werden.

Diese letzte Transformation realisiert die Gebiete hardwaremäßig in der Art eines Paging:

Die pages entsprechen den Gebietsseiten,
die page-frames den Bereichen.

Der Hardwarespeicher ist in Blöcken organisiert. Ein Block ist die einzige adressierbare Speichereinheit und gleichzeitig Transportgröße.

./. 10

2. Gebiete

Wir haben gesehen, daß die Forderung nach Strukturierung der benutzerspezifischen Daten, der Wunsch nach dynamischer Speicherzuteilung und einfacher Speicherabbildung zum symbolisch segmentierten und paginierten Adreßraum geführt hat. (Die Segmente haben wir Gebiete genannt). Jedoch eine Bedingung, nämlich die des unbeschränkten Einschiebens von Information ohne totale Reorganisation des Gebietes (verbunden mit Transporten) ist dadurch noch nicht erfüllt. Diese im Zusammenhang mit der Ablage der Sätze *) einer Datei verständliche Forderung führt dazu, in ein Gebiet zwischen zwei beliebigen Gebietsseiten Gebietsseiten einzufügen oder herauszunehmen.

---

*) Die Ablage der Sätze steht unter zwei Forderungen:

1. Die Zahl der Hintergrundtransporte bei sequentieller Verarbeitung einer mit random (Schreib-)Zugriff aufgebauten Datei soll so groß sein, als wäre die Datei sequentiell aufgebaut.

2. Die Ablage der Sätze einer Datei soll so geschehen, daß das unbegrenzte Einfügen von Sätzen keine totale Reorganisation erfordert.

Forderung 1 bedeutet, daß die mit einem Hintergrundtransport transportierten Sätze einer Datei genau eine zusammenhängende Teilfolge der Sätze umfassen, wobei die Sätze dieser Teilfolge nicht notwendigerweise geordnet sein müssen, da im Kernspeicher random und wortweise zugegriffen werden kann.

Forderung 2 bedeutet, daß im virtuellen Speicher Gebiet Teile eingefügt oder herausgenommen werden können müssen. Denn nur auf diese Weise zieht das unbeschränkte Einfügen in Verbindung mit Forderung 1 keine völlige Umorganisation nach sich. (Da die Betonung auf "unbeschränkt" liegt, fällt auch die Möglichkeit der lückenhaften Ablage fort).

./. 11

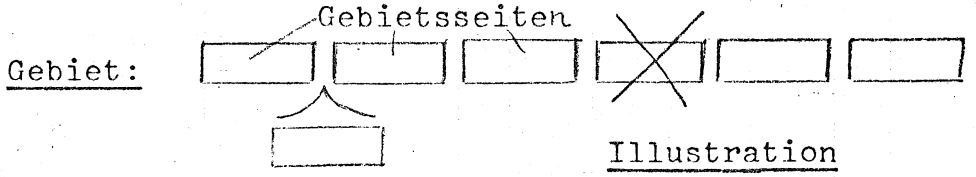
Die Tatsache, daß Gebietsseiten explizit zur Kenntnis genommen und manipuliert werden, verbietet, sie unter dem rein technischen Aspekt der "pages" zu sehen. Andererseits kann man sie aber auch nicht bedenkenlos "Segmente" fester Länge eines linear segmentierten Adreßraums nennen, der Gebiet heißt. Die Gebietsseiten nehmen also eine Zwischenstellung zwischen pages und Segmenten ein.

2.1. Die dynamischen Eigenschaften von Gebieten

1. Ein Gebiet kann kreiert und wieder gelöscht werden. Bei der Kreation wird kein Hardware-Speicher reserviert. Beim Löschen des Gebietes wird der Hardwarespeicher freigegeben.

2. Ein Gebiet kann um beliebig viele Gebietsseiten verlängert oder verkürzt werden. Dabei wird für jede Gebietsseite Speicher von der Größe eines Bereiches reserviert bzw. wieder freigegeben.

3. In die Folge der Gebietsseiten können Gebietsseiten eingeschoben und aus ihr herausgenommen werden, wobei die Adressierung der übrigen Gebietsseiten invariant bleibt. (Diese Gebietseigenschaft realisiert die Forderung nach einer Art "Karteicharakteristik" von Dateien, die besagt, daß uneingeschränktes Einfügen und Streichen von Sätzen ohne umfassende Reorganisation der Datei *) möglich sein muß).



Illustration

*) Die Sätze einer Datei werden so auf die Gebietsseiten verteilt, daß alle Satzmarken der Sätze einer Gebietsseite größer als alle Satzmarken der Sätze voran-

./. 12

2.2. Die Adressierung von Gebieten

Das Paar $(g,\hat{w})$ bezeichnet eine symbolische Adresse $\hat{w}$ des Gebietes g. Beim ersten Transformationsschritt in der Abbildung

symbolische Adresse $\hat{w}$ ⟶ Hardwareadresse

sind die Gebietsseite und die gebietsseitenrelative Adresse zu bestimmen.

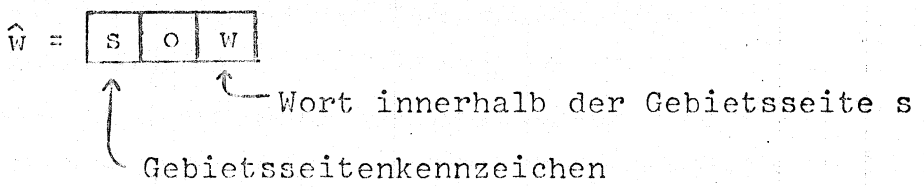
$(\hat{w}) \longrightarrow (s, w)$

s = symbolischer Name der Gebietsseite, Gebietsseitenkennzeichen

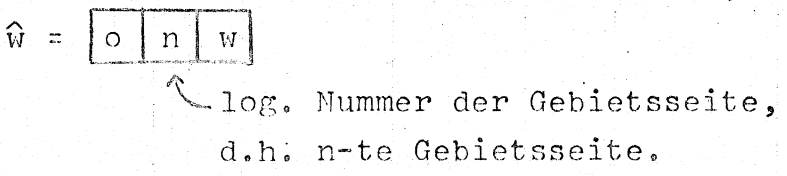
w = Wort innerhalb der Gebietsseite s.

Die symbolische Adresse $\hat{w}$ hat je nach ihrem Aussehen die Bedeutung einer direkten, logischen oder gemischten Adresse.

1. direkte Adressierung



2. logische Adressierung



---

gehender Gebietsseiten und kleiner als alle Satzmarken der Sätze der folgenden Gebietsseiten sind. Bei Überlauf einer Gebietsseite wird eine neue Gebietsseite eingeschoben und es werden die Sätze der überlaufenden Seite auf sie selbst und die eingeschobene Gebietsseite entsprechend der Ordnung der Satzmarken verteilt.

./. 13

3. gemischte Adressierung

$\hat{w}$ = | $\hat{s}$ | $\hat{n}$ | w | , d.h. $\hat{n}$-te Gebietsseite nach der Gebietsseite $\hat{s}$

Man sieht, daß bei direkter Adressierung die Umrechnung $(\hat{w}) \longrightarrow (s, w)$ ablesbar ist, während bei logischer oder gemischter Adressierung die Abbildung berechnet werden muß.

Die direkte und logische Adressierung spiegeln zwei wichtige Verarbeitungsarten eines Gebietes wider:

- wahlfreies Einschieben oder Streichen von Gebietsseiten,

- sequentielle Verarbeitung.

Beim Einschieben und Streichen von Gebietsseiten bleibt die direkte Adressierung invariant, d.h. das Gebiet verhält sich wie eine Kartei, deren Karteikarten, sprich Gebietsseiten, eingefügt und herausgenommen werden können, ohne die Ordnung der übrigen Gebietsseiten zu stören (keine Adressenverschiebung).
Bei sequentieller Verarbeitung des Gebietes (im Sinne der Ordnung der Gebietsseiten) ist die logische Adressierung angemessen. Das Gebiet erscheint bei logischer Adressierung als linearer Adreßraum.

./. 14

2.3. Der Zugriff auf Gebiete

Will man auf Teile eines Gebietes zugreifen, muß man sie vom Hintergrundspeicher in den Kernspeicher bringen. Dabei unterscheiden wir zwei Mechanismen

1. Seiteneinbettung
2. Teilseitentransport

Bei der Seiteneinbettung wird dem Benutzer (hier die über der Gebietsverwaltung liegende Schicht) die verlangte Gebietsseite in einem Puffer zur Verfügung gestellt, auf den er im Sharing zugreift.
Beim Teilseitentransport werden dem Benutzer n konsekutive Teilseiten des Gebietes (konsekutiv unter dem Aspekt des Gebietes als linearer Adreßraum) in seinen Arbeitsspeicher transportiert.

Der wesentliche Unterschied der beiden Zugriffsarten ist, daß bei der Seiteneinbettung die Möglichkeit der Mehrfachbenutzung besteht, dadurch, daß mehrere Benutzer den Puffer im Sharing zugreifen und das System jede Gebietsseite in höchstens einem Puffer hält.
Wenn außerdem das System Sorge dafür trägt, daß eine eingebettete Gebietsseite die aktuelle Version darstellt (d.h. bei Anforderung dieser Gebietsseite bekommt man die schon eingebettete Seite und keine Kopie der ursprünglichen Seite auf dem Hintergrundspeicher) und die eingebettete Seite - wenn nicht mehr zugegriffen wird - automatisch auf den Hintergrundspeicher zurückgeschrieben wird, wird dem Benutzer die Illusion gegeben, er greife direkt auf das Gebiet zu. Im Gegensatz dazu wird dem Benutzer beim Teilseitentransport eine Kopie des Gebietsausschnittes zur Verfügung gestellt.

Seiteneinbettung $\hat{=}$ direkter Zugriff
Teilseitentransport $\hat{=}$ Zugriff auf eine Kopie

./. 15

3. Implementierung

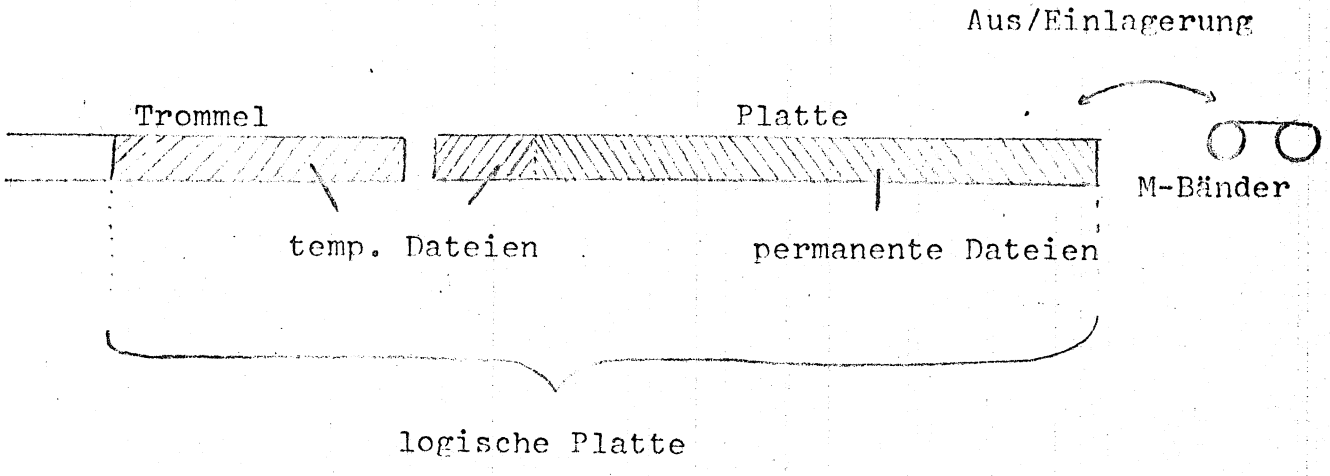
3.1. Die logische Platte

Wir gehen davon aus, daß der Benutzer einer Gesprächssystem-Anlage seine Dateien hinsichtlich ihrer Lebensdauer in zwei Gruppen aufteilt, in

- temporäre Dateien und
- permanente Dateien.

Temporär heißt abschnitts- oder gesprächsgebunden, permanent heißt, über einen Abschnitt oder ein Gespräch hinausgehend, langfristig.

Unter der Annahme, daß auf temporäre Dateien häufiger zugegriffen wird, als auf permanente, werden temporäre Dateien auf der Trommel, permanente und bei Trommelüberlauf auch temporäre Dateien auf der Platte gespeichert. Den kombinierten, aus einem Teil der Trommel und der Platte bestehenden Dateienspeicher nennen wir logische Platte.



3.2. Behälter

Im Folgenden motivieren wir, warum wir den im wesentlichen homogenen virtuellen Speicher, in dem die benutzerspezifischen Daten als Gebiete organisiert abgelegt werden, strukturieren.

1. Die Listen, die Gebiete realisieren, d.h. im wesentlichen die "page-table" des virtuellen Speichers, ist bei einer Plattenkapazität von 30 000K Wörtern etwa 30K lang.

   Wegen ihrer Länge residiert sie auf dem Hintergrundspeicher und wird bei Bedarf teilweise in den Arbeitsspeicher gebracht. Welches sind nun sinnvolle Teile dieser Liste?

   Alle einem Benutzer gehörenden Gebiete bilden eine logisch wie technisch sinnvolle Gruppierungsgröße, die wir Behälter nennen. Ein Behälter ist also eine Organisationsform, die den virtuellen Speicher in benutzerbezogene Teile gliedert.

2. Wie wir später noch sehen werden, gibt es unter dem Aspekt der Verdrängung von Benutzerdaten auf Magnetbänder eine weitere Motivation für die Einführung eines Behälters. Der Behälter erscheint in diesem Zusammenhang als Verdrängungseinheit.

Arten von Behältern

Der Unterschied zwischen temporären und permanenten Gebieten findet seine Entsprechung in permanenten und temporären Behältern. In temporären Behältern sind temporäre Gebiete, in permanenten Behältern permanente Gebiete zusammengefaßt.

./. 17

Somit besitzt jeder Benutzer maximal 2 Behälter, einen temporären und einen permanenten. Während der temporäre Behälter nur für die Dauer eines Gesprächs oder Abschnitts existiert, existiert der permanente Behälter so lange, wie der Benutzer dem System bekannt ist.

3.3. Systemdienste zur Verwaltung von Gebieten

Ein Gebiet g besteht aus einer Folge von Gebietsseiten

$$g = \{s_0, s_1, \ldots, s_k\}$$

wobei s_0 eine virtuelle Gebietsseite ist, deren symbolischer Name systemglobal ist.
(Motivation: Eine virtuelle, "nullte" Gebietsseite erleichtert das logische Adressieren von Gebietsseiten und Einschieben vor der ersten und nach der letzten Gebietsseite).

Parameter
- b = Behälter
- g = Gebiet
- s = Gebietsseitenkennzeichen (= symbolischer Gebietsseitenname)
- n = logische Gebietsseitennummer
- p = Zahl der Gebietsseiten eines Gebietes, die ein Benutzer gleichzeitig im KSP eingebettet haben möchte.

Systemdienste

1. Kreation eines Gebietes ohne Speicherreservierung:

EINRICHTE GEBIET (b)

$\emptyset \longrightarrow g = \{s_0\}$

./. 18

2. Löschen eines Gebietes mit Speicherfreigabe

AUFGEBE GEBIET (b, g)

$g = \{s_o, \ldots, s_k\} \longrightarrow \emptyset$

3. Speicheranforderung (gebietsseitenweise)

ERWEITERE UM GEBIETSSEITE (b, g, s_l), l = o, ..., k

$g = \{s_o, \ldots, s_k\} \longrightarrow g = \{s_o, \ldots, s_l, s_{neu}, s_{l+1}, \ldots, s_k\}$

4. Speicherfreigabe (gebietsseitenweise)

STREICHE GEBIETSSEITE (b, g, s_l), l = 1, ..., k

$g = \{s_o, \ldots, s_k\} \longrightarrow g = \{s_o, \ldots, s_{l-1}, s_{l+1}, \ldots, s_k\}$

5. Aufrufen des Gebietes zur Bearbeitung (Zugriff)
(Motivation: Erst, wenn auf das Gebiet zugegriffen wird, brauchen die Listen, die das Gebiet realisieren, vom Hintergrundspeicher in den Kernspeicher gebracht zu werden)

a) ANMELDE GEBIET MIT PUFFERN (b, g, p)
(für Zugriff durch Seiteneinbettung)

b) ANMELDE GEBIET OHNE PUFFER (b, g)
(für Zugriff durch Teilseitentransport)

6. Beendigung der Bearbeitung eines Gebietes

a) ABMELDE GEBIET MIT PUFFERN (b, g, p)

b) ABMELDE GEBIET OHNE PUFFER (b, g)

7. Zugriff durch Seiteneinbettung

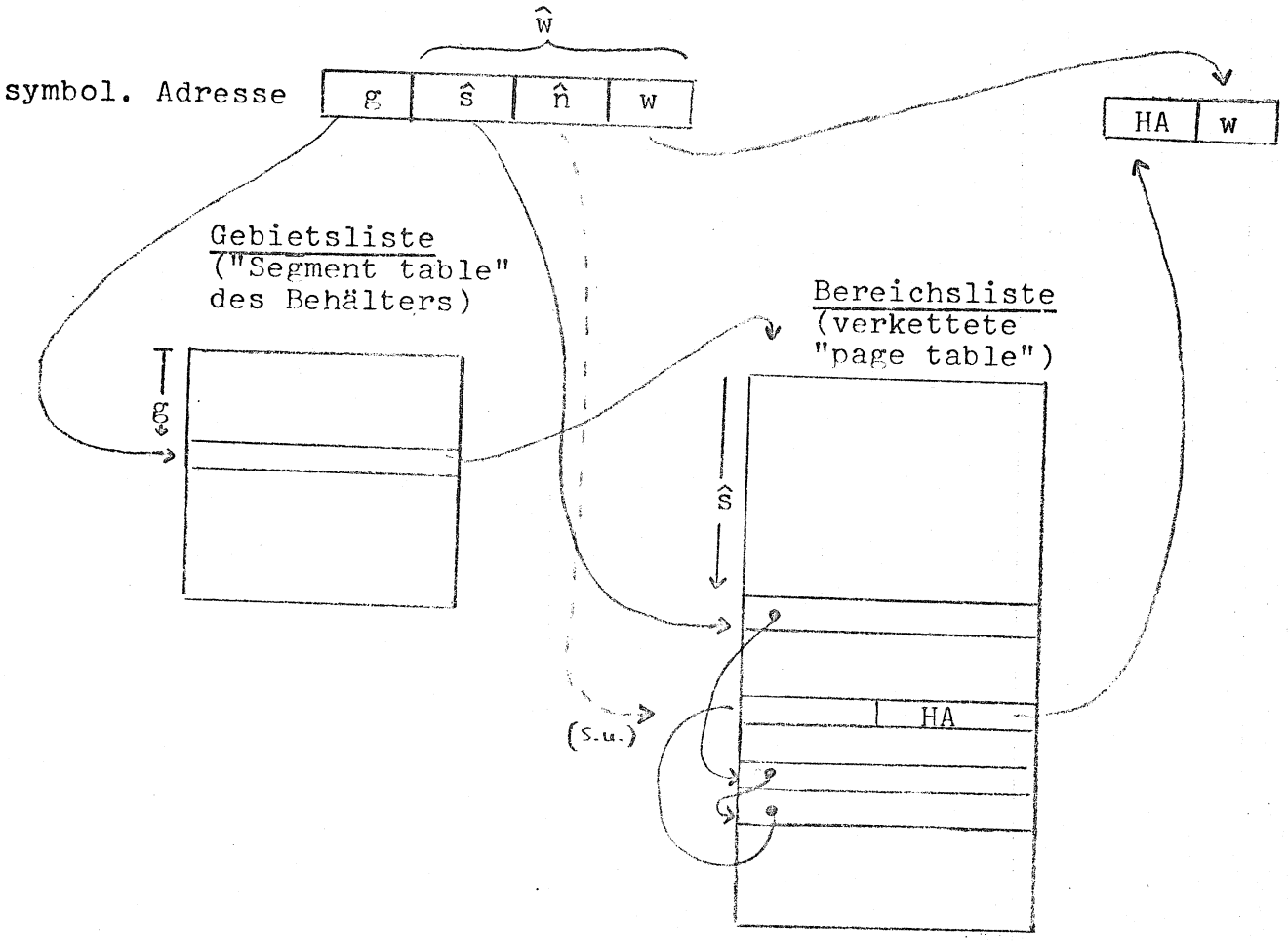
   NIMM UND GIB ZUGRIFF AUF GEBIETSSEITE (b, g, $\hat{s}$, $\hat{n}$)

8. Zugriff durch Teilseitentransport

   a) KOPIERE AUS GEBIET (b, g, Quell-, Zieladresse)

   b) SCHREIBE IN GEBIET (b, g, Quell-, Zieladresse)

3.4. Adreßumsetzung für Gebiete



ŝ ist Einstieg in die Bereichsliste,

n̂ gibt an, im wievielten Element der verketteten Liste vom Einstiegspunkt an gerechnet die gesuchte Gebietsseite steht.

./. 20

3.5. Überlauf des Hintergrundspeichers, Verdrängung

Wegen der endlichen Kapazität der rotierenden Hintergrundspeicher besteht das Problem des Überlaufs. Man kann es dadurch lösen, daß man Informationsmengen, auf die nicht zugegriffen wird, auf andere Speicher verdrängt. An der TR440 Installation wird eine automatische Verdrängung auf Magnetbänder implementiert. Verdrängungseinheit ist der Behälter, weil er

1. die größte *), logische, benutzerspezifische Informationsmenge ist,

2. einen für eine Verdrängung angemessenen Umfang besitzt.

Die Adressen der verdrängten Behälter verzeichnet das System in einer globalen Liste.

Auf die bei Verdrängung auf Magnetbänder und Wiedereinlagerung auf den Hintergrundspeicher entstehenden Probleme der Verdrängungsstrategie, Speicherzuteilung bei Einlagerung und garbage collection auf Magnetbändern soll in diesem Rahmen nicht eingegangen werden.

---

*) wichtig, damit Dateien eines Benutzers nicht auf verschiedene Bänder ausgelagert werden.

4. Verfeinerung des Gebietskonzepts

Im folgenden wird das Gebietskonzept im Hinblick auf bessere Speicherausnutzung modifiziert.
Bei der Abbildung des virtuellen Speichers, der durch die Organisationsform GEBIET realisiert wird, auf den physikalischen Speicher stehen zwei Forderungen im Zielkonflikt:

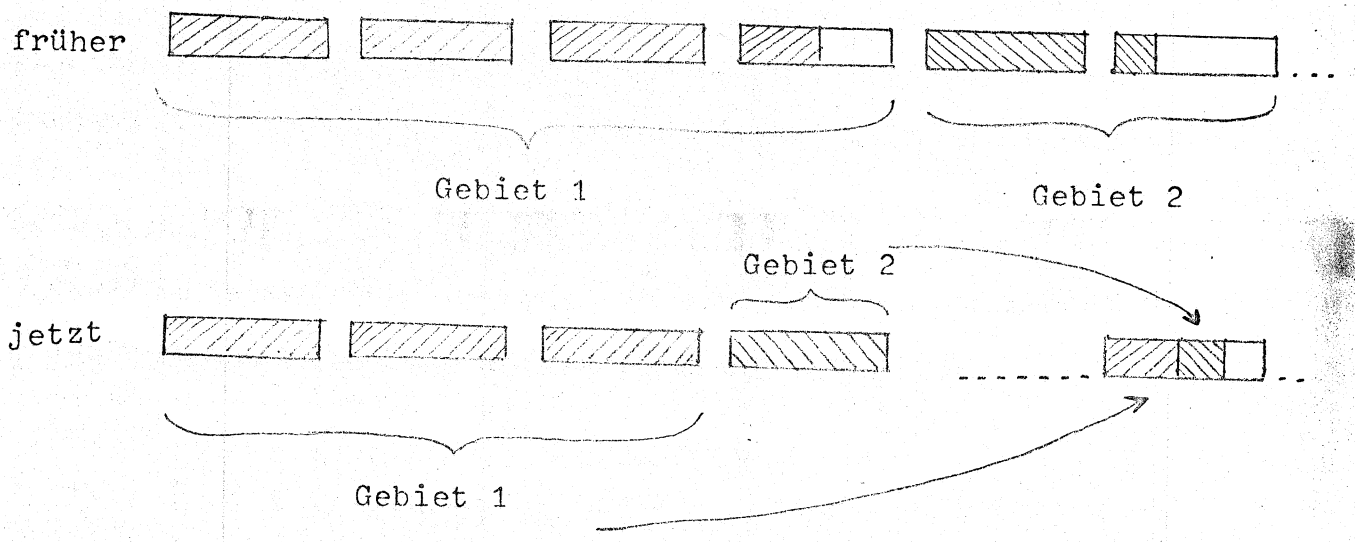
1. Einfachheit der Speicherzuteilung

2. Geringhalten des Speicherverschnitts und des Verwaltungsoverheads.

Da gemäß dem Schichtenbild der eigentliche Übergang vom virtuellen Speicher zum Hardwarespeicher in einer tiefen Schicht des Systems liegt und die Systemphilosophie auf hardwarenäheren Schichten einfache und effiziente Algorithmen bevorzugt, liegt es nahe, die erste Forderung auf einer tieferen Schicht, die zweite auf einer höheren Schicht, d.h. Speicherebene zu erfüllen.
Auf der untersten Speicherebene bleibt es bei der uniformen Speicherzuteilungseinheit, die wir als Bereiche eingeführt haben.
Auf der Gebiets-Speicherebene geschieht die Speicherzuteilung in zwei verschieden großen Einheiten, deren eine wie bisher die Gebietsseite, die andere die Teilseite ist. Eine Teilseite ist eine n-tel Gebietsseite.

Die Verwaltung der Teilseiten geschieht benutzerspezifisch, d.h. auf Behälterebene aus Gründen der behälterweisen Verdrängung (Vermeidung zu starker Splitterung des Behälters).

./. 22



Die Speicherzuteilung auf Gebiets-Speicherebene geschieht teilseitenweise, wobei n Teilseiten zu einer Gebietsseite zusammengefaßt und als solche verwaltet werden. D.h. ein Gebiet besteht im wesentlichen wie früher aus Gebietsseiten, nur an seinem Ende aus Teilseiten. Dadurch wird der Speicherverschnitt von ursprünglich einer halben Gebietsseite pro Gebiet auf eine halbe Teilseite pro Gebiet (plus eine halbe Gebietsseite pro Behälter) verringert.
Die Zuteilung unterschiedlich großer Adreßraumstücke zur besseren Speicherausnutzung wird mit einem höheren Verwaltungsauswand erkauft. Der trade-off zwischen Verwaltungsaufwand und kompakter Informationsablage muß aber unter folgendem, systemspezifischen Aspekt gesehen werden:

./. 23

Für die hierarchische Verwaltung der benutzerspezifischen Daten kennt das System Datenbasen, das sind Organisationseinheiten mit Verweisinformation, die in sehr kleinen Gebieten von der Größe einer Achtelseite liegen. Außerdem gibt es vergleichbar kleine Gebiete für die Ablage der Indexhierarchie zum schnelleren Wiederauffinden von Sätzen einer random-sequentiellen Datei. Für diese relativ zahlreichen Gebiete wäre ein Speicherverschnitt von über 80 % - wie er beim alten Gebietskonzept vorliegt - nicht zu verantworten. Deswegen wird das Gebietskonzept hinsichtlich der Zuteilung virtuellen Speichers in Teilseiten der Größe einer Achtelgebietsseite erweitert.

Zusammenfassung:

Die Speicherzuteilung geschieht auf zwei Ebenen:

1. Auf der physikalischen Ebene in uniformen Zuteilungseinheiten, den Bereichen.
   (Motivation: Einfachheit, keine äußere Speicherfragmentierung).

2. Auf der virtuellen Speicherebene in zwei verschieden großen Zuteilungseinheiten, den Gebietsseiten und den Teilseiten.
   (Motivation: Verringerung des inneren Speicherverschnitts bei Unterbringung von Objekten zweier Größenkategorien:

   1. Gebiete der Größe mehrerer Gebietsseiten (große Gebiete, vorwiegend mit Benutzerinformation).

   2. Gebiete der Größe weit unterhalb einer Gebietsseite (kleine Gebiete, vorwiegend mit Datenverwaltungsinformation).