UNIVERSITÄT LEIPZIG

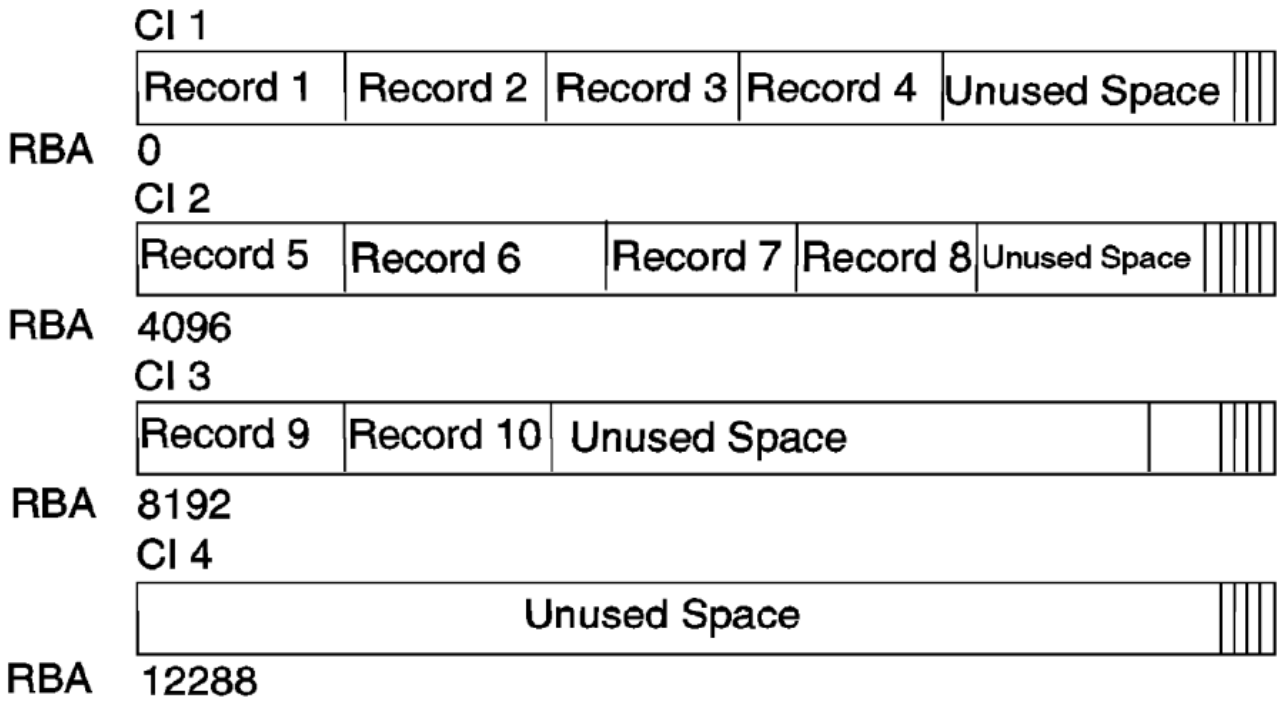
Enterprise Computing
Einführung in das Betriebssystem z/OS

Prof. Dr. Martin Bogdan
Prof. Dr.-Ing. Wilhelm G. Spruth

WS2012/2013

Datasets Teil 3

VSAM Dataset Organisation – ESDS und KSDS

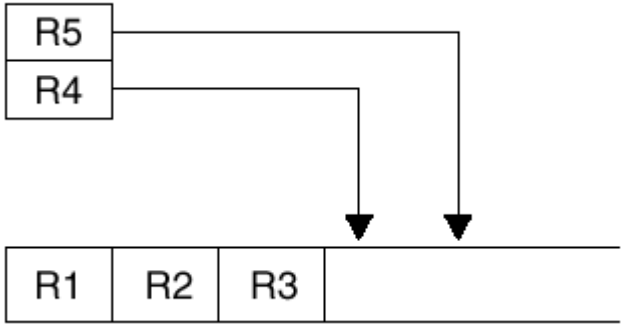


Entry Sequenced Dataset (ESDS)

Ein ESDS ist mit einem sequentiellen Dataset vergleichbar. Er enthält Records mit fester oder variabler Länge. Records werden in der Reihenfolge ihres Eintreffens in dem Dataset abgespeichert, nicht auf Grund des Wertes in einem Schlüsselfeld. Alle neuen Datensätze werden am Ende des Datasets gespeichert.

Dargestellt ist ein ESDS mit 4 Control Intervallen (CI 1, CI 2, CI 3, CI 4), die jeweils 4096 Bytes lang sind. Der Dataset enthält 10 Datensätze und einigen ungenutzte freien Speicherplatz. Der Offset jedes CI vom Anfang des Datasets wird als relative Byte-Adresse (RBA) bezeichnet. Die 4 CIs beginnen bei den relativen Byte-Adressen (RBA) 0, 4096, 8192 und 12228, gezählt vom Anfang des Datasets.

Records werden in der Reihenfolge ihres Eintreffens in dem Dataset abgespeichert, nicht auf Grund des Wertes in einem Schlüsselfeld.



Neue Records werden im Anschluss an die bisher gespeicherten Records abgespeichert. Vorhandene Records können nicht gelöscht werden. Wenn Sie einen Record löschen möchten, müssen Sie diesen Record als inaktiv kennzeichnen. Soweit es VSAM betrifft, wird der Record nicht gelöscht. Records können aktualisiert werden, aber sie können nicht verlängert werden. Um die Länge eines Records in einem ESDS zu vergrößern, müssen Sie ihn am Ende des Datasets als neuer Record speichern.

Für einen ESDS werden zwei Arten von READ-Verarbeitung unterstützt:

- **Sequentieller Zugriff (die häufigste).**
- **Direkter (oder zufälliger) Zugang. Dies erfordert, dass das Anwendungsprogramm die Relative Byte-Adresse (RBA) des Records benutzt. Die RBA muss irgendwie dem Programm bekannt sein.**

Typische sequentielle Lesevorgänge in einem Anwendungsprogramm benutzen Befehle wie GET NEXT (Datasetname, ...). Dies lädt denjenigen Record aus dem Bufferpool, der dem zuletzt abgerufenen Record folgt.

Typical ESDS processing

Zusätzlich zum sequentiellen Zugriff erlaubt ein ESDS einen direkten (random) Zugriff. Hierzu muss das Anwendungsprogramm die relative Byte-Adresse (RBA) des gewünschten Records angegeben.

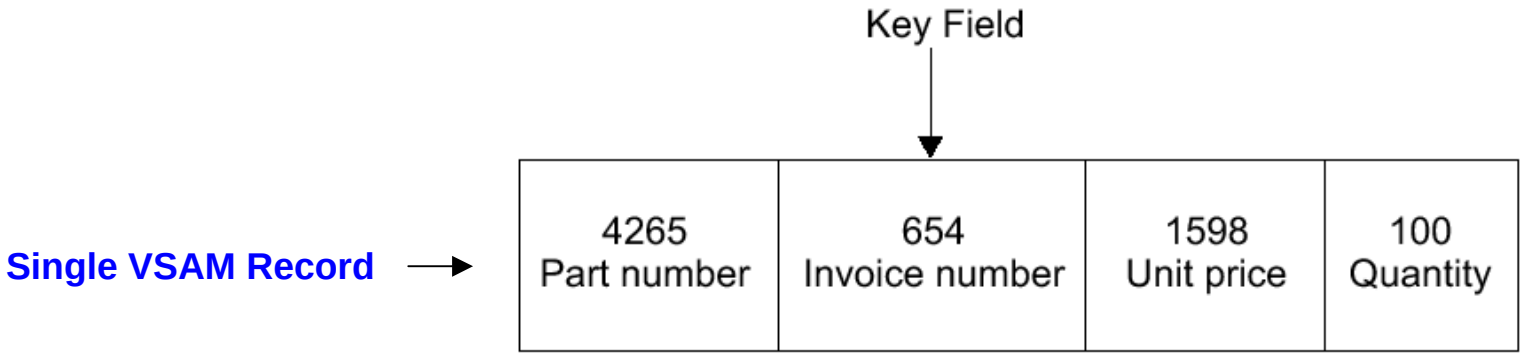
Wenn ein Record geladen oder hinzugefügt wird, zeigt VSAM seine relative Byte-Adresse (RBA) an. Der RBA ist ein Zeiger, der das Offset (displacement) in Bytes dieses Recorda vom Anfang des Datasets enthält. Der erste Record in einem Dataset hat den RBA Wert 0. Der Wert des RBA für den zweiten (und nachfolgenden Records) hängt von der Länge der bisherigen Records ab..

Der RBA eines logischen Records hängt nur von der Position des Records innerhalb des Datasets ab. Der RBA wird immer als Fullword binäre ganze Zahl ausgedrückt.

Beispiel von RBAs eines Entry-Sequenced Datasets
X’62’ = Decimal 98

Key Sequenced Dataset (KSDS)

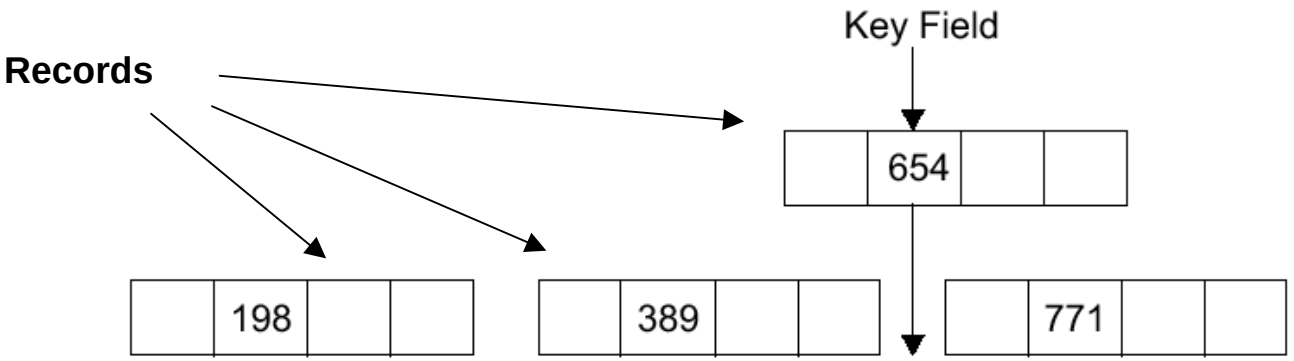
Jeder KSDS Record besteht aus mehreren Feldern. Eines dieser Felder kann als Schlüsselfeld (Key Field) deklariert werden.



In diesem Beispiel besteht jeder VSAM Record aus 4 Feldern. Die Rechnungsnummer wird als Schlüssel-Feld verwendet. Für den gezeigten Record ist der Wert des Schlüsselfeldes 654.

Das Schlüsselfeld muss sich in der gleichen Position in jedem Record eines Key Sequenced Datasets befinden (das zweite Feld in der Abbildung). Der Offset des Schlüsselfeldes vom Anfang des Records und die Schlüssellänge sind benutzerdefiniert. Der Schlüssel jedes Records muss eindeutig sein. Nachdem ein Record in einen VSAM Dataset geladen wurde kann der Wert des Schlüssels nicht mehr verändert werden. Der gesamte Record muss gelöscht und neu geschrieben werden.

In einem Key Sequenced Dataset werden logische Datensätze in aufsteigender Schlüsselreihenfolge entsprechend des Wertes in dem Schlüsselfeld platziert. Der Schlüssel enthält einen eindeutigen Wert, wie z.B. eine Personalnummer oder Rechnungsnummer. Dies bestimmt die Sortierreihenfolge (Collating Sequence) der Records in dem KSDS Dataset. Für einen bestimmten KSDS wird die Schlüssellänge auf einen konstanten Wert im Bereich von 1 bis 255 Bytes festgelegt. VSAM unterstützt keine Schlüssel mit variabler Länge innerhalb eines einzigen KSDS. Das Schlüsselfeld wird als ein binärer Wert behandelt.



Wenn Records aktualisiert werden, ihre Länge verändert wird, neue Records eingefügt werden, Records gelöscht werden, werden alle Records mit höheren Schlüssel-Werten innerhalb des Control Intervalls verschoben.

In dem gezeigten Beispiel enthält der VSAM KSDS drei Records mit den Schlüssel Werten 198, 389 und 771. Ein neuer Record mit dem Schlüssel Wert 654 wird nach Record 389 und vor Record 771 eingeordnet. Das Schlüsselfeld der Records bestimmt die Reihenfolge, in der die Records gespeichert werden.

In einem KSDS können Records sowohl sequentiell als auch direkt (indexed) abgearbeitet werden, entsprechend ihren Schlüsselwerten. Die Vorteile des KSDS sind:

- **Sequentielle Verarbeitung ist zum Abrufen von Datensätzen in der sortierten Reihenfolge nützlich**
- **Die direkte Verarbeitung von Records ist bei on-line-Anwendungen nützlich.**

Collating Sequence

Der Begriff Collating Sequence bezieht sich auf die Reihenfolge, in der Zeichenfolgen platziert werden, wenn sie sortiert werden sollen.

Ein typisches Beispiel ist die bekannte "alphabetische Reihenfolge", in der "Alfred" vor "Zeus" auftritt, weil „A" vor "Z" im Alphabet erscheint. In einem Computer-System treten jedoch zusätzliche Reihenfolge Probleme auf:

- **Groß-und Kleinschreibung: ein Großbuchstabe "A" und ein kleines "a" werden in der Regel als der gleiche Buchstabe betrachtet. So erscheint Ab zwischen AA und AC. Aber es kann sein, dass Sie die Records anders sortieren möchten.**
- **Nationale Sonderzeichen, Akzente, Tilden: Verschiedene Sprachen verwenden diese Marken über und um Buchstaben herum, aber ein Sprecher mag diese Buchstaben gleich behandeln.**

In einem Computersystem, wird zwangsläufig jedem Buchstaben einen einzigartiger numerischer Code (wie in ASCII, EBCDIC oder Unicode-Zeichensätzen) zugewiesen. Die ordnungsgemäße und übliche Anordnung von Zeichenketten entspricht nicht einen einfachen numerischen Vergleich dieser Zeichensatz Codes. Vielmehr wird die Reihenfolge anhand der Sortierreihenfolge (Collating Sequence) bestimmt.

In der deutschen Sprache werden Umlaute (Ä, Ö, Ü) in der Regel ebenso wie ihre nicht-Umlaut Versionen behandelt. Der Buchstabe ß wird immer als ss sortiert. Dies ergibt die alphabetische Reihenfolge Arg, ärgerlich, Arm, Assistent, Aßlar, Assoziation. Für Telefon-Verzeichnisse und ähnliche Listen von Namen werden die Umlaute wie die Buchstaben-Kombinationen "ae", "oe", "ue" behandelt. Dies ergibt die alphabetische Reihenfolge Udet, Übelacker, Uell, Ülle, Ueve, Uffenbach.

Die Sortierfolge in einer EBCDIC Umgebung ist nicht das Gleiche wie die Sortierfolge in einen ASCII-Umgebung. VSAM-Schlüssel werden nach der EBCDIC Sortierfolge sortiert.

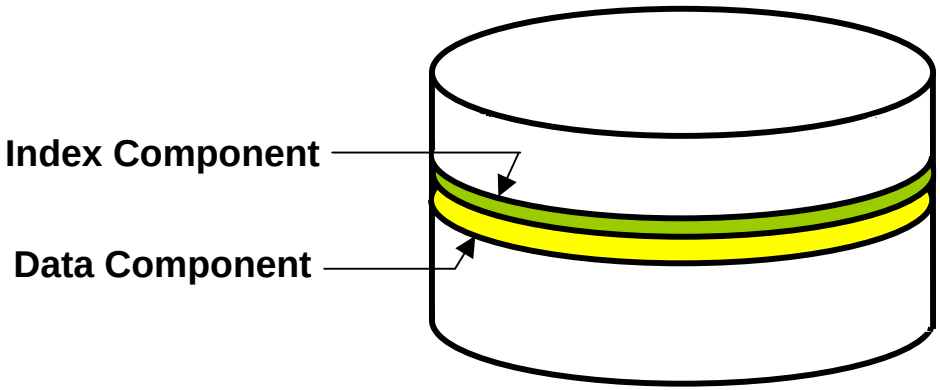
Before

After

In der Abbildung oben sind zwei logische Records in dem oberen Kontroll-Intervall (CI) gespeichert. Die beiden Records haben die Schlüssel 11 und 14. Das untere Kontroll-Intervall zeigt, was passiert, wenn Sie einen Record mit einem Schlüssel von 12 einfügen..

1. **Record 12 wird in seiner korrekten Sortierfolge in dem CI eingesetzt.**
2. **Das CI-Definition Feld (CIDF) wird aktualisiert, um die Reduzierung des verfügbaren freien Speicherplatzes zu registrieren.**
3. **Ein entsprechendes Record Definition Field (RDF) wird an der entsprechenden Stelle eingefügt, um die Länge des neuen Records zu beschreiben.**

Wenn ein Record gelöscht wird, wird der Vorgang umgekehrt durchgeführt. Der Speicherplatz des gelöschten Records und des entsprechenden RDF wird als freier Speicherplatz zurückgewonnen.

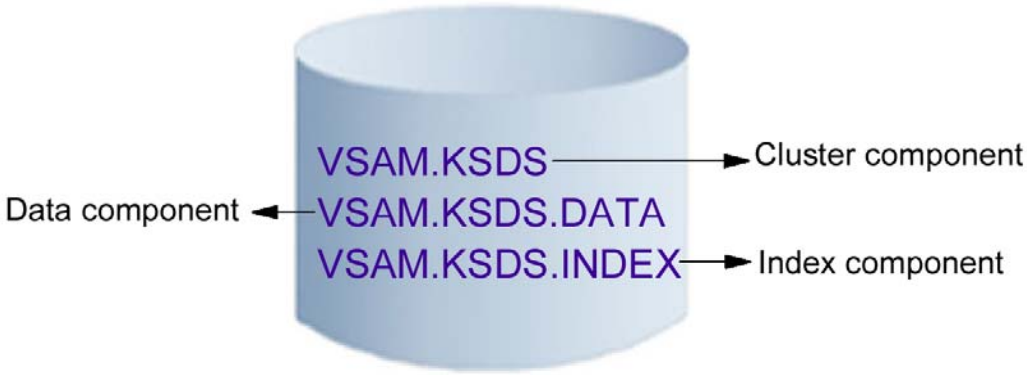


VSAM Key Sequenced Dataset (KSDS)

Auf einem Plattenspeicher nimmt ein Key Sequenced Dataset (KSDS) zwei lineare Speicherbereiche ein, sogenannt Komponenten. Es gibt zwei Arten von Komponenten, die Daten-Komponente und die Index-Komponente. Die Daten Komponente ist der Teil einer VSAM-Datei, welcher die Daten Records enthält. Alle VSAM Datasets haben eine Daten-Komponente. Andere VSAM Organisationen, zum Beispiel Entry Sequenced Datasets (ESDS), haben nur die Daten-Komponente.

- **Die Data Component ist der Teil des VSAM Datasets der die (Daten) Records enthält. Alle VSAM Datasets beinhalten eine Datenkomponente.**
- **Die Index Component ist eine Sammlung von Records (Index logical Records), welche**
    - **Data Keys der Records der Datenkomponente enthalten, sowie deren**
    - **Adressen (RBA, Relative Byte Address).**

Unter Verwendung des Indexes (Index Component) ist VSAM in der Lage, einen logischen Datensatz aus der Datenkomponente abzurufen, wenn eine Anforderung nach einem Datensatz mit einem bestimmten Schlüssel erfolgt.



Cluster

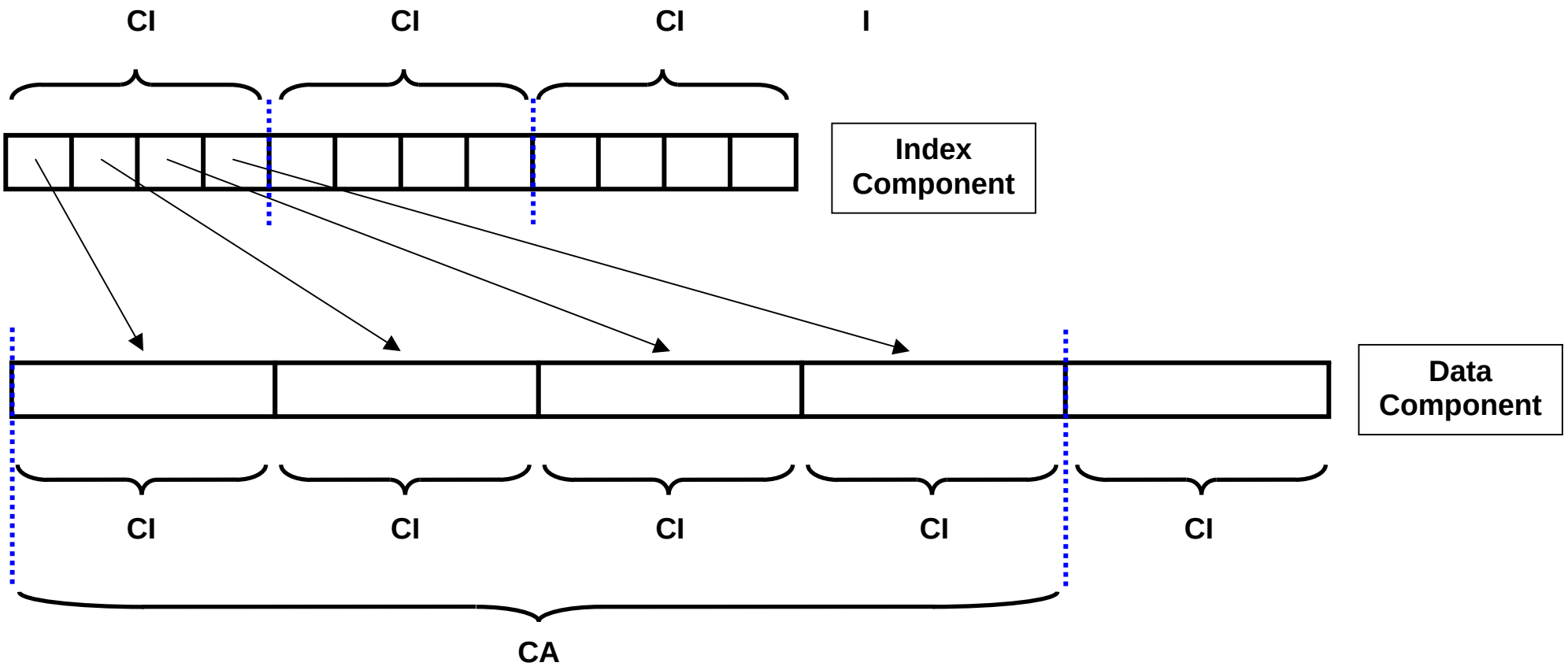
Die Datenkomponente und die Index-Komponente sind wirklich zwei unabhängige VSAM Datasets.

Ein VSAM-Cluster ist die Kombination der Daten Komponente (Dataset) und der Index-Komponente (Dataset) eines KSDS. Das Cluster Konzept vereinfacht die VSAM Verarbeitung. Es bietet eine Möglichkeit, Index und Daten-Komponenten als eine Einheit zu behandeln, und mit einem eigenen Namen zu katalogisieren. Es kann auch jede Komponente einen eigenen Namen haben. Dies ermöglicht es, die Daten Komponente getrennt von der Index-Komponente zu verarbeiten.

Jetzt kommt der entscheidende VSAM Frage: "Was in VSAM entspricht einem z/OS Dataset?" Die beste Antwort ist, es hängt von den Umständen ab. Wenn Sie zum Beispiel den Cluster-Namen in einer DD-Anweisung eines JCL-Script benutzen, dann entspricht der Cluster dem Dataset. Wenn Sie sich auf eine Komponente beziehen, dann entspricht diese dem Dataset.

(Die Bedeutung des Begriffs "Cluster", im Kontext mit VSAM, ist nicht identisch mit der Bedeutung des Begriffs "Cluster" in einer parallel Processing Configuration.)

In unserer VSAM Übungsaufgabe definieren wir einen VSAM Cluster mit dem Namen PRAKT20.VSAM.STUDENT, der aus einer Data Component PRAKT20.VSAM.STUDENT.DATA sowie einer Index Component PRAKT20.VSAM.STUDENT.INDEX besteht.



Die Index-Komponente besteht aus mehreren CIs (und wahrscheinlich auch aus mehreren CAs). Jeder CI in der Index-Komponente besteht aus mehreren Index Records, wobei jeder Index Record in der Index-Komponente auf ein CI in der Daten Komponente zeigt.

In dem oben gezeigten Beispiel enthält jedes CI in der Index-Komponente 4 Records, und jede CA in der Data Component enthält 4 CIs.

Multilevel Index Component

Ein VSAM-Index (Index Component) kann aus einer einzigen Ebene oder aus mehr als einer Ebene bestehen. Jede Ebene enthält Zeiger auf die nächst tiefere Ebene.

Die Suche in einem großen Index kann sehr zeitaufwendig sein. Ein mehrstufiger Index wird aufrechterhalten, um die Index-Suche zu verkürzen. VSAM teilt die Index Komponente in zwei Teile auf: Sequence-Set und Index-Set. Die unterste Ebene der Index Komponente wird als Sequence-Set bezeichnet. Die Pointer in der untersten Ebene zeigen direkt (mit Hilfe einer RBA) auf ein Kontroll-Intervall (CI) innerhalb einer Control Area (CA) der Daten Komponente.

Der Sequence Set enthält

- **einen Index-Eintrag für jedes CI in der Daten Komponente, und damit auch**
- **ein Index CI für jedes CA in der Daten-Komponente.**

Bei kleinen VSAM Datasets würde die Index Komponente nur aus einem Sequence Set bestehen. Größere Index Sets unterhalten als Bestandteil ihrer Index Komponente eine oder mehrere Index Ebenen zusätzlich zu dem Sequence Set. Man bezeichnet die Ebenen oberhalb des Sequence Set als den Index Set. Er kann beliebig viele Ebenen enthalten. Sequence-Set und Index Set bilden zusammen die Index-Komponente eines KSDS.

Ein Eintrag in einem Index Set Datensatz enthält den höchstmöglichen Key in einem Index-Datensatz in der nächst tieferen Ebene, und einen Zeiger auf den Anfang dieses Index-Datensatzes. Die höchste Ebene des Index enthält immer einen einzelnen Index CI.

In der in dem folgenden Beispiel gezeigten Abbildung besteht die Daten Komponente eines VSAM Key Sequenced Dataset (KSDS) aus 3 Control Areas. Jede CA besteht aus 6 Kontrollintervallen, und jedes CI speichert genau 3 Records. Es kann nützlich sein, wenn Sie die Seite mit dem Beispiel ausdrucken um den folgenden Text besser zu verstehen.

Der Sequence Set in der Index-Komponente enthält 3 CIs, eine CI für jede CA in der Daten Komponente. Jedes CI in dem Sequence Set enthält 6 Records, einen Datensatz für jedes CI in der Daten-Komponente. Jeder Sequence Set Record enthält den höchsten Key in dem zugehörigen Daten-Komponente CI.

Die erste CI in der Daten-Komponente hat die Schlüssel 2, 5 und 7. Der erste Datensatz in dem Sequence-Set enthält einen Zeiger (RBA) an den Datensatz mit dem Schlüssel 7 in der Daten-Komponente.

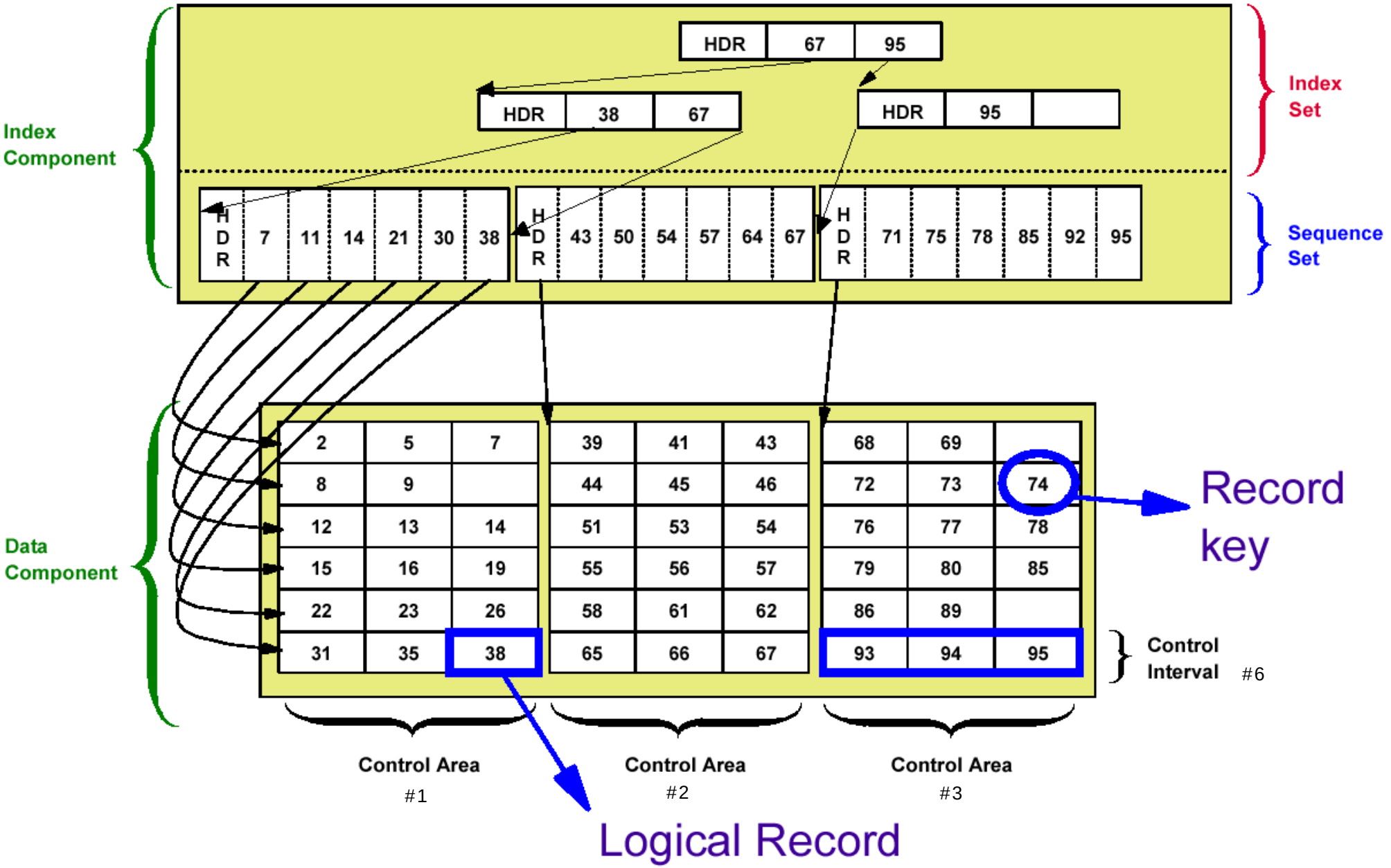
Das zweite CI in der Daten-Komponente hat die Keys 8, 9, und einen freien Platz. Das dritte CI in der Daten-Komponente hat die Keys 12, 13 und 14. Wenn der freie Speicherplatz in dem zweiten CI in der Daten Komponente später gefüllt wird, kann ihr Key einen Wert nicht höher als 11 haben (oder es wäre an anderer Stelle platziert werden). Deshalb enthält der zweite Datensatz in dem Sequence Set einen Zeiger (RBA) auf einen potentiellen Datensatz mit dem Schlüssel 11 in der Daten-Komponente.

Für die zweite CA in der Daten-Komponente enthält der Sequence Set eine neue CI. Der erste Datensatz des CI enthält den Wert 43, weil das der Schlüssel des letzten Datensatzes in CI Nr. 1 von CA Nr. 2 in der Daten-Komponente ist.

Das letzte CI in der ersten CA der Datenkomponente hat die Keys 31, 36 und 38. Die Index Ebene unmittelbar über dem Sequence-Set enthält einen Indexeintrag für jedes Sequence Set Controll-Intervall. Damit enthält der erste Datensatz in der ersten CI des Index Set den Wert 38. Der zweite Datensatz in dem ersten CI des Index Set enthält den Wert 67. Dies ist der höchste Key in CA 2 der Daten-Komponente.

Die unterste Ebene des Index Set enthält 2 CIs, jeweils ein CI für die beiden CAs 1 und 2 der Daten-Komponente, und ein weitere CI für CA 3 der Daten-Komponente. Diese beiden CIs werden durch eine zweite Ebene des Index Sets adressiert.

Alle Einträge werden in aufsteigender Key Reihenfolge gehalten.


Index Component
HDR 67 95
HDR 38 67
HDR 95
Index Set
HDR 7 11 14 21 30 38
HDR 43 50 54 57 64 67
HDR 71 75 78 85 92 95
Sequence Set
Data Component
2 5 7
8 9
12 13 14
15 16 19
22 23 26
31 35 38
39 41 43
44 45 46
51 53 54
55 56 57
58 61 62
65 66 67
68 69
72 73 74
76 77 78
79 80 85
86 89
93 94 95
Record key
Control Interval #6
Control Area #1
Control Area #2
Control Area #3
Logical Record

Es existieren zwei weitere VSAM Dataset Formate, die nicht so häufig wie die VSAM Enty Sequenced (ESDS) und VSAM Key Sequenced (KSDS) Dataset Formate eingesetzt werden:

Relative Record Dataset (RRDS)

Records in einem RRDS werden in Slots mit einer festen Länge geladen. Eine weitere Version mit einer variablen Slot Länge hat den Namen Variable-Length RRDS (VRRDS). In beiden Fällen werden die Records durch die Relative Record Numbers (RRNs) ihrer Slots adressiert.

Ein Verarbeitungsprogramm benutzt RRNs für einen direkten (random) Zugriff auf die Records.

Linear Dataset (LDS)

Ein LDS-Set ist ein VSAM Dataset mit einem Control Interval Größe von 4096 Byte bis 32768 Byte. Ein LDS enthält nur eine zusammenhängende Kette von Datenbytes. Er hat keine Control Informationen in seinem CI, das heißt, keine Record Definition Felder und kein Control Intervall Definition Feld (CIDF). Daher sind alle LDS Bytes Datenbytes. Wenn der LDS logischen Records enthält, müssen diese durch das Anwendungsprogramm geblocked und entblocked werden. Records existieren nicht aus der Sicht von VSAM. So weit erfolgt die Verarbeitung ähnlich wie in einem Unix Hierarchical File System.

Ein LDS wird für spezielle Anwendungen benutzt, die es erfordern, große Datenmengen im Hauptspeicher zu halten.

Zusammenfassung

Ein Control Interval ist die Einheit der Daten, die zwischen Festplattenspeicher und virtuellen Speicher übertragen wird, wenn eine I/O-Anforderung auftritt. Es enthält Records, freien Speicherplatz und Steuerinformationen.

Eine Gruppe von Control Intervals bildet eine Control Area (CA). Eine typische Größe ist ein Zylinder eines IBM Modell 3390 Plattenlaufwerks.

Der Index ist ein Merkmal eines KSDS. Ein Sequence Set ist der Teil des Index, der auf die Control Area und das Control Interval eines Records verweist. Der Index Set ist der andere Teil des Index. Es hat mehrere Ebenen mit Zeigern, die letztendlich auf den Sequence Set zeigen.

Ein Control Intervall SPLIT ist die Übertragung von Records von einem CI in ein neues CI, um Platz zu schaffen. Das Ergebnis sind zwei halb leere CIs. Ein Control Intervall Split erfordert eine Reihe von I/O Operationen, was CPU Performance kostet. Um dies zu verringern sollte immer ausreichend freier Platz in den CIs vorgesehen werden.