

RECHENZENTRUM TH MÜNCHEN
ARBEITSGRUPPE BETRIEBSYSTEME

INTERNSCHRIFT Nr. 25

THEMA:

Die Anmeldung von Files, die von mehreren V's oder P's gemeinsam benutzt werden

VERFASSER:

Wolf

DATUM:

28.4.1969

FORM DER ABFASSUNG

SACHLICHE VERBINDLICHKEIT

ENTWURF
 AUSARBEITUNG
 ENDFORM

ALLGEMEINE INFORMATION
DISKUSSIONSGRUNDLAGE
ERARBEITETER VORSCHLAG
VERBINDLICHE MITTEILUNG
VERALTET

ÄNDERUNGSZUSTAND

BEZUG AUF BISHERIGE INTERNSCHRIFTEN

Vorkenntnisse aus:

Erweiterung von:

Ersatz für:

BEZUG AUF KÜNFSTIGE INTERNSCHRIFTEN

Vorkenntnisse zu:

Erweiterung in:

Ersetzt durch:

ANDERWEITIGE LITERATUR

Die Anmeldung von Files, die von mehreren V's oder P'sGemeinsam benutzt werden1 Vorbemerkungen:

- 1.1 Will ein P oder V mit einem File arbeiten, der auf einem Hintergrundspeicher liegt, so muß dieser File ganz (Fall a) oder teilweise (Fall b) in den Kernspeicher gebracht werden.
- a) Der File wird ganz in den Kernspeicher gebracht (sofern möglich); Mit dieser im Kernspeicher stehenden Kopie des Files wird gearbeitet; u.U. wird diese Kopie geändert und das Original auf dem Hintergrundspeicher mit der geänderten Kopie überschrieben. Nun ergibt sich folgendes Problem: Für einen gewissen Zeitraum existieren zwei (evtl. voneinander abweichende) Exemplare des Files. Welches ist das "Referenzexemplar", d.h. welches Exemplar wird zur Verfügung gestellt, wenn der File ein zweites Mal verlängt wird? Man vergleiche dazu einen analogen Sachverhalt beim TR4: Ist ein Programm bereits einmal gestartet worden, so steht es sowohl im Kernspeicher als auch auf Band. Soll das Programm erneut gestartet werden und ist es ausdrücklich als "nicht wiederstartbar" gekennzeichnet, so wird es neu vom Band geladen (Referenzexemplar auf Band). Fehlt eine solche Kennzeichnung, so wird (Zeitersparnis!) das bereits im Kernspeicher stehende Exemplar des Programms benutzt; das setzt jedoch voraus, daß das Programm tatsächlich wiederstartbar ist, d.h. sich nicht in unzulässiger Weise selbst verändert hat. Bereits aus diesem Beispiel erkennt man, daß etwa Programm-Files Einschränkungen unterworfen werden müssen, wenn man jeweils das Kernspeicherexemplar eines solchen Files als Referenzexemplar betrachten will.

b) Es werden nur einzelne Teile (Seiten) des Files in den Kernspeicher gebracht. Dort können sie evtl. geändert und später wieder zurückgespeichert werden. Hier ist klar, daß als Referenzexemplar nur der File auf dem Hintergrundspeicher in Frage kommt.

1.2 In den Fällen a) und b) taucht die Frage auf: Kann während des Zurückschreibens des Files bzw. der geänderten Teile auf den File zugegriffen werden? Dies kann nicht erlaubt werden, da die im File auf dem Hintergrundspeicher enthaltene Information während des Zurückschreibens inkonsistent sein kann. (Analog: Wird durch einen Befehl der Inhalt einer Speicherzelle verändert, so darf während der Ausführungsphase des Befehls nicht auf diese Zelle zugegriffen werden.)

1.3 Für die folgenden Überlegungen werde nun vereinbart:
Als "Referenzexemplar" eines Files wird stets das Exemplar auf dem Hintergrundspeicher betrachtet. Auch der Begriff "Zugriff" wird (wenn nicht ausdrücklich anders vermerkt) auf dieses Exemplar bezogen. Während des Überschreibens von Teilen des Files auf dem Hintergrundspeicher ist der Zugriff gesperrt.

2 In welchen Modi kann auf einen File zugegriffen werden?

2.1 Es sollen hier kurz folgende Möglichkeiten diskutiert werden:

Möglichkeit A (Multics):

- A1) Lesen
- A2) Schreiben
- A3) Ausführen
- A4) Information anhängen

Möglichkeit B (TR 440):

- B1) Lesen (einschließlich Ausführen)
- B2) Schreiben (einschließlich Information anhängen)

Eine andere Formulierung von B ist:

Möglichkeit C:

- C1) Zugriff ohne Veränderung des Files
- C2) Zugriff mit Veränderung des Files

2.2 Zu diesen Möglichkeiten ist anzumerken:

2.2.1 Es ist anzunehmen, daß alle (sinnvollen) Einteilungen in Modi Verfeinerungen der Einteilung C sind.

2.2.2 Zur Aufteilung des Modus C1 (Zugriff ohne Veränderung) in die Modi A1 (Lesen) und A3 (Ausführen):

In beiden Modi wird der File oder einzelne Seiten des Files in den Kernspeicher gebracht und nicht wieder zurückgespeichert. Vom hier eingenommenen Standpunkt aus (Bezug auf das Exemplar auf dem Hintergrundspeicher) erscheint die Aufteilung nicht unbedingt notwendig.

2.2.3 Für die Aufteilung des Modus C2 (Zugriff mit Veränderung) in die Modi A2 (Schreiben) und A4 (Information anhängen) können folgende Argumente geltend gemacht werden:

a) Der Modus A4 kann (im Gegensatz zu A2) die Länge des Files derart verändern, daß die Zuteilung zusätzlichen Hintergrundspeichers erforderlich wird. z.B. kann der ursprünglich zugewiesene Speicherbereich überlaufen, wenn an eine Liste eine sehr große Anzahl neuer Elemente angefügt werden soll.

b) Es sind Fälle denkbar, bei denen das Lesen in einem File erlaubt werden kann, auch wenn eine Änderung des Files im Modus A4 ausgeführt werden soll: Es ist z.B. sinnvoll, einen Benutzer eine Bibliotheksliste zur Verfügung zu stellen, die konsistent, wenn auch nicht auf dem neuesten

Stand ist, anstatt diese Anfrage bis nach dem Abschluß der Änderung im Modus A4 zurückzustellen. In anderen Fällen wird der Zugriff gesperrt werden müssen, wenn ein File nicht konsistent ist, ehe die Änderung im Modus A4 abgeschlossen ist, z.B. Verweise enthalten, die ins Leere führen.

3 Wie kann sichergestellt werden, daß auf einen File von 1 P oder V korrekt zugegriffen wird (entsprechend dem Inhalt des Files und der Absicht des File-Eigentümers)?

3.1 Hier sind z.B. folgende Möglichkeiten denkbar:

3.1.1 Bei jedem File ist angegeben, in welchen Modi und von welchen Benutzern auf diesen File zugegriffen werden darf. In Multics ist diese Methode noch verfeinert: Bei jedem Filenamen (ein File kann mehrere Namen haben, je nachdem, über welche Verweistabellen man sich auf ihn bezieht) steht ein Zugriffsbit für jeden Modus (A1, A2, A3, A4) und evtl. ein Verweis auf eine Liste von Benutzern. Ist dieses Bit gesetzt, so ist der Zugang in diesem Modus für die in dieser Liste eingetragenen Benutzer erlaubt. Ist dieses Bit nicht gesetzt, so ist der Zugriff in diesem Modus verboten.

3.1.2 Es existiert ein Filekatalog, in dem bei jedem eingetragenen File spezifiziert ist, welche Benutzer in welchen Modi zugreifen dürfen.

3.1.3 Bei jedem Prozeß ist eingetragen, auf welche Files er zugreifen darf (TR 440).

Diese Möglichkeit kann etwa dahingehend abgeändert werden, daß z.B. der Zugriff auf alle Files eines Kataloges (die z.B. eine bestimmte Bibliothek enthalten) erlaubt wird.

3.1.4 Weiterhin ist eine Anzahl weiterer Kombinationen und Abwandlungen der angegebenen Methoden denkbar. Die einzelnen Lösungen dürften sich jedoch u.U. nicht unerheblich hinsichtlich Speicherbedarf

und Zeitbedarf (für das Durchsuchen von Listen) unterscheiden.

3.2 Anmerkungen über den Zugriff vom System aus:

Die Teile des Betriebssystems können einer Zugriffskontrolle nur begrenzt unterworfen sein:

- 3.2.1 Die (Hardware-) Lese- und Schreibroutinen müssen freien Zugang zu jedem File haben.
- 3.2.2 Die Fehlerbehandlungs- und Verdrängungsroutinen sollten nicht nur ungehinderten Zugang zu jedem File haben, sondern auch (falls nötig und evtl. mit Meldung an den Eigentümer des Files) die Zugangsberechtigung ändern können (Zugriffsbits umsetzen).
- 3.2.3 Andere Teile des Systems sollten sich (im allgemeinen) hinsichtlich des Zugangs wie Benutzerprogramme verhalten.

4 2 oder mehr unabhängige, einander unbekannte P's greifen auf einen File zu

4.1 Folgende Fälle sind denkbar:

- a) n P's lesen (Modus A1);
- b) n P's führen aus (Modus A3);
- c) 1 P schreibt (A2) und mindestens 1 P liest (A1) bzw. führt aus (A3).
- d) 1 P hängt Information an (A4) und mindestens 1 P liest (A1) bzw. führt aus (A3);
- e) 2 (oder mehr) P's schreiben (A2);
- f) 2 (oder mehr) P's hängen Information an (A4).

4.2 Dazu ist zu bemerken:

Die Fälle a) und b) können ohne Bedenken gestattet werden. Die Fälle c) und e) sind im allgemeinen zu verbieten, desgleichen Fall f), da während einer Änderung des Files durch Anhängen von Information der Bezugspunkt für eine zweite solche Änderung fehlt. Für den Fall d) vergleiche man die bereits unter 2.2.3 (zur Aufteilung von C2 in A2 und A4) gemachte Anmerkung.

5 Zwei einander bekannte P's greifen auf einen File zu

Zwei P's seien etwa Parallelzweige eines Programms (z.B. in Algol 68, PL/I). Es wird angenommen, daß die beiden P's untereinander die korrekte Synchronisation selbst besorgen; die übliche Zugriffskontrolle soll ausgeschaltet werden, andere P's sollen am Zugriff gehindert werden.

5.1 Das könnte folgendermaßen realisiert werden:

Es wird die Möglichkeit geschaffen (etwa mittels eines zusätzlichen Zugriffsbits C), daß der Zugriff entweder wie in 3. beschrieben erfolgt (C aus) oder nur mittels Codewort (ohne Rücksicht auf gesetzte oder nicht gesetzte Zugriffsbits). Dabei sollte mit dem Codewort der Name des betreffenden Prozesses bzw. Benutzers verbunden werden, um Verwechslungen zu vermeiden.

5.1.1 Es seien alle P's gleichberechtigt:

Es kann angenommen werden, daß die Anmeldungen von Files sequentiell abgearbeitet werden. Sind nun P2 und P3 Parallelzweige eines Programms, so hat jeder der beiden zunächst anzufragen, ob der File bereits angemeldet ist; falls nicht, muß er den File unter Angabe des Codeworts anmelden. Ist der File bereits angemeldet, so ist der Zugang nur bei Angabe des richtigen Codeworts gestattet.

5.1.2 Für Gruppen von P's seien "Leit-P's" ausgezeichnet:

Das Leit-P P1 spalte in zwei Unter-P's P2 und P3 auf:
P1 hat dann das Codewort an P2 und P3 weiterzugeben
und über den Zugriff zum gemeinsamen File Buch zu führen.

5.2 Eine andere Möglichkeit zur Organisation der Parallelarbeit wäre die folgende:

Die Semaphors, die von den einzelnen P's benutzt werden,
liegen im Systemspeicher. Werden von 2 P's Semaphors
gleichen Namens benutzt, so erkennt das System, daß
diese P's zusammenarbeiten wollen und überwacht die
Parallelarbeit.

6 Ergänzung zur Internschrift Nr. 23

- 6.1 Die Internschrift Nr. 23 befaßt sich mit der Anmeldung von Files, die von mehreren P's oder V's gemeinsam benutzt werden. Dort wird festgestellt, daß es im allgemeinen nicht erlaubt werden kann, daß ein P oder V in einem File schreibt, während andere P's oder V's in diesem File lesen.
- 6.2 Dieses Verbot kann unter folgenden einschränkenden Voraussetzungen aufgehoben werden (ICL, Betriebssystem George):
 - 6.2.1 Für den File wird festgelegt, daß nur serieller Zugriff (serial access) erlaubt ist.
 - 6.2.2 Jedem P oder V wird ein "Zeiger" zugeordnet, der angibt, wie weit das betreffende P oder V in diesem File geschrieben bzw. gelesen hat.
 - 6.2.3 Das System kontrolliert, daß der Zeiger des schreibenden P's oder V's nicht vom Zeiger eines der lesenden P's oder V's überholt wird.