2312ZenginSystemDown
2023.12 のブログ：「全銀システム障害の原因について」、の詳細
（→ http://www.1968start.com/M/blog/index3.html#2312 ）

全銀システム障害の原因について

中所武司

■このブログのきっかけ

日経と朝日の記事によると、10 月 10 日に発生した全銀システム障害について、
全銀ネットと NTT データから説明があったとのこと。ソフトウェア工学の視点で検討してみた。

＜参考記事１＞ 日経の記事（2023.12.1）
「全銀システム障害の真因はメモリーの確保領域不足、全銀ネットと NTT データが発表」
https://xtech.nikkei.com/atcl/nxt/news/18/16364/

＜参考記事２＞ 朝日の記事（2023.12.2）
「全銀ネット障害は設定ミス 管理不十分認める 影響５５０万件」
https://digital.asahi.com/articles/DA3S15807129.html

＜参考記事３＞ 日経の記事（2023.12.4）
「全銀システム障害「詳細設計書見落とし」でオーバーフローの痛恨、再発防止なるか」
https://xtech.nikkei.com/atcl/nxt/column/18/00001/08680/

■記事内容とコメント（→★）

●**＜参考記事１＞** 日経の記事（2023.12.1）
「全銀システム障害の真因はメモリーの確保領域不足、全銀ネットと NTT データが発表」

・全国銀行資金決済ネットワーク（**全銀ネット**）と NTT データは 12 月 1 日、
「全国銀行データ通信システム（**全銀システム**）」のシステム障害に関して会見。
障害の原因は、OS のバージョンアップに伴うテーブルサイズ拡張の考慮漏れにより、
作業領域が不足したことだった。

・全銀システムと金融機関のシステムをつなぐ中継コンピューター（RC）の OS 更改に伴い、
金融機関名テーブルのサイズを拡張したが、確保すべきメモリー領域が不足したことが真因。

・商用環境の共有メモリー上にある金融機関名などに関するテーブルを、ディスクエリアの
ロードファイルから展開する生成プログラムにおいて、確保する作業領域の不足が発生。
ロードファイルが**一部破損**し、商用環境の共有メモリー上のテーブルにも破損が生じた。

→★この表現から推測すると、全銀システムで使用する**金融機関情報テーブル**は、
ディスク内にロードファイルとして保存されており、全銀システム稼働時に、
その生成プログラムが、**作業領域**を経由して、共有メモリーにコピーするようだ。

→★この場合の生成プログラムの処理内容を推測すると、
ロードファイル全体を一度には読み込まず、先頭から少しずつ複数回に分けて読み込む。

その１～N 回分の読み込んだデータを、生成プログラムの作業エリアに書き込んだ後、
それを共有メモリーのしかるべき場所に書き出す、という処理を繰り返し、
最終的にロードファイルのすべてを読み込み、共有メモリーに書き出したら処理終了とする。

→★このとき、「作業領域の不足が発生」とのことなので、
作業領域を超えて金融機関情報テーブルを書き込んだ後、
作業領域のデータのみ、共有メモリーのしかるべき場所に書き出したか、あるいは、
作業領域を超えて書き込まれた部分が共有メモリーに書き出す前に変更されて、
「共有メモリー上のテーブルにも破損が生じた」ということかな。

→★OS のバージョンアップに伴い、テーブルサイズを拡張したにもかかわらず、
生成プログラムの作業領域を増やさなかったのは、初歩的ミスと言える。

・全銀システムの障害は 10 月 10 日に発生。10 の金融機関で、他行宛て振り込みに関する
オンライン処理が 2 日間にわたりできなくなった。処理の遅延など何らかの影響を受けた
振込処理は、仕向け（送信側）と被仕向け（受信側）を合わせて 566 万件に上った。

・全銀ネットと開発会社は、11 月 30 日に発生原因の分析や再発防止策などを金融庁へ報告した。

●＜参考記事２（日経との重複部分は省略）＞ 朝日の記事（2023.12.2）
「全銀ネット障害は設定ミス 管理不十分認める 影響５５０万件」

・全国銀行資金決済ネットワーク（全銀ネット）の銀行間送金システムの障害の原因は、
更新したプログラムの設定ミスだった。設計や試験プロセスに問題があり、
全銀ネットも管理が不十分だった。事前のテストも不十分だった。

→★開発側の最終工程でのシステムテスト、および、全銀側の受入テストにおける、
テスト仕様書、テスト項目・テストケースをみてみたいものだ。
今回のミスが事前のテストで検出できなかったのは、ソフトウェア工学の観点で言えば、
よほど、「**テストケースが網羅的でなかった**」と思われる。

→★まさかとは思うが、今回はバージョンアップ作業だったので、
更新したプログラムに関連すると思われるテスト項目だけを抜粋して実施した可能性もある。
ソフトウェア工学的観点では、更新プログラムによる他への悪影響の有無の確認のため、
広範囲の再テスト（リグレッションテスト、回帰テスト）が必要だった。

＜参考：拙著「ソフトウェア工学」（朝倉書店）の「7.4 テスト手順」から引用＞
http://www.1968start.com/M/lecture/SE3index.html

『累積情報はプログラムが変更されると無効になり，過去に実行したテストを再びやりなおす
回帰テスト（regression testing）が必要になる．そこで，各テストケース単位に
テストデータ入力から結果の確認までをテスト手続きとして記述することにより，
テスト実行を自動化することが有効である』

→★過去の「必要な領域を確保していなかった」ケースでは、領域のオーバーフローがある。
今回も作業領域のオーバーフローのチェック機能（数行のプログラム）があれば防げた。
ただし、今回のミスは、これより初歩的ミスと思われる。

【過去ブログ】
＊2016.2　「オーバーフローのチェック漏れがなくならない」
http://www.1968start.com/M/blog/old.html#1602
＊2015.6　「オーバーフローのチェック漏れがなくならない」
http://www.1968start.com/M/blog/old.html#1506c

・開発会社の社長は「システム上の直接の原因は当社の責任」と謝罪。
システムの設計や製造、試験のプロセスを改めるとした。

→★開発部門とは独立して、ユーザ視点で品質保証（検査）を実施する部門の有無が気になる。

・一方、全銀ネットは、訓練や人材が不足していたため、委託元としての管理が不十分で、
復旧対応にも影響したと説明した。利用者に対する補償は１１月１７日時点で約８千件、
計約８００万円といい負担は今後詰める。

→★まともな受け入れテストの実施の有無、運用マニュアルにおける事故対応の記述の有無
などが気になる。

●＜参考記事３（上記の日経・朝日との重複部分は省略）＞ 日経の記事（2023.12.4）
「全銀システム障害「詳細設計書見落とし」でオーバーフローの痛恨、再発防止なるか」

（注）有料会員限定のため、無料公開の１ページ目のみ閲覧］

・全国銀行データ通信システム（全銀システム）のシステム障害に関する 12 月 1 日の会見で、
以下の 2 つの重要な事実が新たに判明した。

＊各金融機関から他行宛ての送金時に中継コンピューター（RC）が参照する
金融機関名テーブルとそのテーブルの参照を高速化する 3 種類のインデックステーブルが
メモリーへの展開時、所定の作業領域からオーバーフローして破損したこと。

＊詳細設計書では 4 種類のテーブルを同時に展開できる作業領域の確保を求めていたが、
プログラマーやレビュアーなどの関係者がいずれもその記述を見落としたこと。

・今回の障害の直接的な原因は、金融機関名テーブルを生成するプログラムが
必要な作業領域を確保できていなかったことだ。

・全銀ネットは、14 の金融機関で RC を RC17 シリーズから RC23 シリーズへ更改した。
RC23 では、OS を従来の 32 ビットから 64 ビットに変更した。
さらに、OS が Linux 系であること、生成プログラムの開発言語が C 言語であること、
金融機関名テーブルが long 型の属性値を保有していたことを明らかにした。

・long 型は 32 ビット版は 4 バイト、64 ビット版は 8 バイトのメモリーが必要なので、
メモリー増が必要となり、サイズ拡張後の金融機関名テーブルが確保したメモリー内に
収まると確認していた。

・しかし、金融機関名テーブルと同時に 3 種類のインデックステーブルを展開する仕様だった。
インデックステーブルは、1132 の金融機関の金融機関名テーブルの参照を高速化するもの。

・しかし担当者はこれを１つずつ展開すると誤認した結果、確保したメモリーを超えた領域に
展開されたデータが他のプログラムから上書きされ、異常値が紛れ込んだ。

→★この誤認は信じがたい。「1 つずつ展開する」と誤認した場合、作業領域の使用管理に関し、
使用可能領域のポインターの初期化処理、展開前のこのポインターの参照、
展開後のこのポインターの変更処理などが必要のはず。
そして、展開前には、空き領域が十分あるか否かのオーバーフローチェック処理があるはず。

・4 つのテーブルを同時に展開する設計については詳細設計書には記載されていたものの、
製造工程で見落とされ、コーディングでもレビューでもすり抜けた。

→★さらに、システムテストも、品質保証部（検査部）も、受入テストもすり抜けた。

以上