

Page5 UCW と UCB の生成について

UCB(OS でデバイスを代表するコントロールブロック)は、IPL/NIP 中に作成されます。IODF は LOADPARM か LOADxx により指定されます。

(真ん中の HCD DYNAMIC を見ましょう)

ここはハードウェア、つまり CHIPID, CU (Control Unit), デバイスの追加、削除が可能です。同様に、UCB と EDT (Eligible Device Table: 装置グループ) についても編集が可能です。これらは上のセントラルストレージ上に置かれます。HSA (UCW) と OS から見えるメインストレージ上に UCB を作ります。

(UCW はサブチャンネルが使うデバイス装置を示すコントロール領域)

(次に右の HCD を見てください)

ここは LPAR の名前、チャンネルの情報、CU, デバイス情報が UCW と UCB, EDT と NIP コンソールに反映されます。HCD はコンフィグレーションの変更をし、IODF を扱います。

(最後に左側の SE です)

IOCDS を選ぶことで、UCW イメージを選びます。IOCDS は IOCP により書き込まれます。

Page10 ハードウェア IPL 概要

- 処理の実際は、Z/Architecture に依存しています
- ハードウェアにより処理されます
- ひとつの CPU が IPL に専有されます。他のプロセッサはマニュアルステート(停止状態)に置かれます。
- プロセス開始前に一旦、リセットがかかります。(これはプロセッサより IO 装置にとって重要です)

- ICKDSFにより、IPL レコード(ブートプログラム)はディスクにかかれます。
 - シリンダー0、トラック0、レコード1、2、および IEAIPL00 レコードです。

Page12 ハードウェア IPL のサマリー

- ハードウェアはアドレス0番地に、IPL テキスト(24バイト)を読み込む CCW を作ります。
 - CCW = 02000000,40000018
 - DASD の場合、常に cylinder0,track0,record1 を読みます。
- アドレス8番地は、続きの CCW として解釈されます。
 - record2 をストレージに読み込む。コマンドチェーンされた次の CCW です。
 - record2 が示すアドレスの CCW が実行されます。(実行 CCW の転送)
 - IEAIPL00 レコードをシークし、サーチします。(さらに続く CCW の実行によって)
 - アドレス0番地に、IEAIPL00 を読み込みます。
- 一連の CCW チェーンが完結すると、絶対アドレス0番地に置かれたデータを PSW に読み込み、実行を開始する。
 - IEAIPL00 のロケーション0が最初の PSW 内容

※訳者注:

CCW には Transfer in Channel というコマンドがある。これは次に実行する CCW のアドレスが示された CCW で、I/O そのものを行うわけではない。チャンネルプログラムにおけるジャンプ機能のようなもの。

Page13 IPL 資源の初期化

✚ 概要

- もともとは、Nucleus をロードし、マスターアドレススペースを作るだけだった。
 - XA アーキテクチャとダイナミック I/O のサポートにより、複雑になってきた。
- 処理はシングルスレッド。
- IPL ベクターテーブル(IVT)は、このフェーズにおける全体に関する情報をもつ。

✚ IEAIPL00

- 最低限の OS。(再配置不可)

- 最初の仮想記憶環境を作る。
 - IPL 作業領域は、仮想アドレス x'20000000'に置かれる。
- サービスが提供されるのは、
 - リアルフレームをもった仮想アドレス
 - I/O
- ・IPL 初期化のプロセスをコントロール。
 - リソース初期化モジュール(RIM)を IPL 作業領域にロード。
 - コントロールを渡す(実行する)。

Page14 IPL RIM のプロセス

1. Test Block(TB)命令の実行。(ストレージをクリアする)
2. SCPINFO を読む。
 - loadparm を入手。
 - autostore status をオンにする。
3. 使えるリアルストレージに、メモリーの最高位にあわせる。
4. IPL ロードパラメータを入手し、さまざまなデフォルト値をセット。
5. LOADxx を探し、LOADxx 内の情報を処理。
6. IODF を探し、IODF 内の情報を処理。
 - NIP コンソールテーブルを作成。
 - IPL RIM によりサポートされる NIP コンソールの最大数は 64。(HCD では 128)
 - 追加情報は APAR OA12877 を参照。

Page15 IPL RIM のプロセス(続き)

6. (続き)

- デバイスの UIM を探す。
 - デバイス固有の Nucleus と LPA モジュールを決定。
 - SQA と ESQA の必要量を計算。
 - 作業領域にデバイスコントロールブロックを作成。
 - アロケーションのための EDT を作業領域内に作成。

7. DAT-on Nucleus の CSECT マップを作成。

- IEANUC0x と IEANUC2x のモジュールを含む、これらは NMLx,NUCLSTxx,UIM により識別される。
- CSECT は、属性と RMODE と Read-Only かにより、グルーピングされ配置される。

8. モジュールをロードし、外部参照を解決する。

Page16 IPL RIM のプロセス(続き)

9. SQA/ESQA の作成

- IBM 提供値、LOADxx INITSQA, UIM などの合計値で決まる。

10. マスターの VSM コントロールブロックと LSQA を作る。

11. マスターのパーマネントページとセグメントテーブルを作る。

12. 作業域から SQA/ESQA に移す。

- Device Control Block
- アロケーション用 EDT
- IPL メッセージ
- LPA デバイスサポートモジュールリスト

13. リアルストレージの検証をし、使用可能なフレームのキューを作る。

- IPL 作業域はここで破棄。

14. プレフィックス・レジスターをロード。

15. PSA の Nucleus ヘスイッチ。

注:これらは IPL RIM の概略である。IPCS で見ることのできる、IPL 統計報告書に出る IPL RIM の完全なリストは、このレポートの終わりに掲載した。

Page18 LOADxx 探索順序

LOADPARAM にかかれた LOADxx メンバーを探す順序。xx は 5 ケタ目 6 ケタ目からなる。(例:Load Parm=012355M)

Page19 NIP 資源初期化

概略

- 基本的なシステム・リソースの初期化。
- 処理はマルチスレッドで行われる - 通常のディスパッチは動作している。
- 基本的なシステム・サービス (SRB, WAIT, POST, EXCP, ATTACH など) は初めに利用可能になる。
- 追加のサービスは、NIP RIM の実行によって有効となる。
- NIP ベクターテーブル (NVT) は、このフェーズにおける全体に関する情報をもつ。

コントロールルーチン

- 予期せぬエラーへのトラップを用意。(まだ RTM はない)
- ハードウェア環境の検証。
- IPL プロセッサ・コントロールブロックを作る。
- グローバル VSM (仮想記憶管理) のコントロールブロックを作る。
- I/O コントロールブロックのプールを作る。
- 初期システムトレーステーブルを作る。
- LNKLST として SYS1.NUCLEUS をオープン。
- NIP RIM ルーチンをロードして開始。

MVS がデバイスを扱うためには、

- デバイスの UCW がなければいけません
- デバイスの UCB がなければなりません

デバイスをマッピングする過程で

- マッチした UCW が使用可能とされます
- マッチした UCB が接続されます

マッピング処理の過程で、POR(または Dynamic I/O の更新)に伴い HSA にロードされる I/O コンフィグレーション(UCW)は、IODF によって定義される OS の構成情報(UCB)と一致します。※POR:Power On Reset

UCW は POR 直後やシステムリセットによって、利用不能の状態となる。

UCB の初期状態は、

- UCB は、"not connected"(未接続)状態のビット(UCB byte7 の bit2)が、1(オン)となって作られる。
 - マッピングプロセスがすべて終わると、チャンネルサブシステム(UCW)と MVS(UCB)の両方に定義されたすべてのデバイスは、利用可能で接続された状態となる。
 - 適切な UCB のない UCW は利用不能のままとなる。
 - 適当な UCW のない UCB は未接続状態のままとなる。
- いずれかの状態にあるデバイスは、システムによって使われることはない。

■ パスの使用可能性は、パスを調査して行われる。

- このプロセスで MVS はデバイスに対して生成されているすべてのパスがオンラインになるかチェックする。それはデバイスに繋がったパスを指定して I/O オペレーションを試みることで試す。

- 最低限、1 つでもパスがあればデバイスはオンラインになる。
- テープは例外である。パスはオフラインのテープデバイスにも行われる。
MVS は、パスを調べている間、どのパスやデバイスについても使用可能かどうかのレポートはしない。

Page22 DASD デバイスのパス

- DASD パスの調査中、オンラインにならないパスがあればオペレータに問い合わせるので、NIP コンソールは DASD パスを調べる前に利用可能になる必要がある。
 - SSCH をマルチパスに対して投げる(並行してマルチパスデバイスを調べる)。
 - それぞれの I/O が成功したら、他のデバイスをテストする。
 - 該当デバイスへの I/O が終了したら、他のデバイスへかける(コントローラには複数デバイスが下がっている)。
 - 15 秒以内に I/O が完了しない場合、I/O オペレーションはページされる。
 - それぞれのパスのテストをする。
 - 1.5 秒のタイムアウトにならないこと(メッセージ IOS120A は、テストの間は出ない)。
 - IOCDs 内の CHIPD を表す PIM(Path Installed Mask)を作る。
 - I/O につかう UCB 内のコントロールパスとして使われる LPM(Logical Path Mask)を作る。
 - Device Characteristics を求める。ひとつのパスから(情報取得のための CCW を)投げる。
 - デバイス自身による自己記述情報がそれぞれのパスからあがる(ダメな場合は、メッセージ IOS291I)。
 - VOLSER のチェック。Self-describing Product のものはひとつのパス、そうでなければすべてのパスを試す。
 - 重複ボリューム名のメッセージ。(SYSRES でなければ IEA213A、SYSRES なら IEA214A)
 - 最後に 15 秒間、仕掛かり I/O が終わるのを待つ。
 - 今回ダメだった UCB にはマークをつける。後ほどテストが行われる。
 - すべての仕掛かり中の I/O リクエストをページする。

Page23 DASD デバイスのパス(続き)

DASD のパス検査は4つのフェーズをもつ：

それぞれのパスを試す(P)、Read Device Characteristics を出す(D)、自身の情報提示(S)、VOLSER 処理(V)。

DASD パス設定の間に検知されたすべてのエラーは NIP コンソールに出力される。

メッセージは IGGN504A、IGGN505A、IEC334I、IOS291I、IEA213A、IEA214A(末尾の A は、オペレータ応答の必要なメッセージであることを示す)

CCW=チャンネルコマンドワード

SDP=Self-describing Product

RCD=Read Configuration Data

SSID=Subsystem ID

RDC=Read Device Characteristics

Page24 DASD デバイスのパス(続き)

■ パス検査中のエラーメッセージ

- ・ IOS291I メッセージ(RC=21,27,29)はコンフィグレーションエラーである可能性を示す。
- ・ DASD CU のコンフィグレーション誤りによって、重複 SSID を示すエラーメッセージが IOS291I に追加されることもあります。
- ・ パス検査の最終ステップが、DAVV(Direct Access Volume Verification)です。
- ・ DAVV はオンライン状態の DASD のボリュームラベルを読み、UCB に VOLSER を設定します。
- ・ 重複 VOLSER がある場合、IEA213A または IEA214A メッセージが表示されます。

Page25 DASD デバイスのパス(続き)

■ パス検査中のエラーメッセージ

- ・ デバイスがビジー状態であれば IOS120A メッセージがやがて表示される。
- ・ IPL を完了させるために必要なデータセットが、パス検査中にビジー状態とされたボリュームにあると、IGGN504A、IGGN505A メッセージが出力される。
- ・ 共有されている DASD の VOLSER に対応したデバイス番号を得るために、アクティブなシステムから「D U,VOL=vvvvv」コマンドを発行する。
(訳者注:IGGN505A メッセージへのオペレータ対応として、代替 UNIT 番号を知るために別システムから D U コマンドを使え、と言う例示)

1. RTM リカバリーとコントロールブロックを作る。
2. WTO コントロールブロックとプールを作る。
 - WTO はここから SYSLOG に書き出される。
3. マシンチェックハンドラー(MCH)の初期化。
4. UCW から UCB へのマッピングが使用可能か試し、DASD でないデバイスを初期化する。
5. NIP の初期化。
 - WTO はここから NIP コンソールに出る。
6. DASD に対して使用可能かを試し、初期化する。(DASD パス調査)
 - オペレータは検査の間であることを示される。
7. マスターカタログのオープン。
8. IEASYMxx からシステム・シンボルを作る。
9. SVCLIB、PARMLIB および LOGREC をオープン。
10. 必要ならシステム・パラメーターの入力プロンプトを表示。(メッセージ IEA101A)
11. システム・パラメーターのマージと解析。

12. ASM(Auxuary Storage Manager)の初期化、ページとスワップデータセットのオープン。
13. SQA=パラメータの処理。
 - クイックスタート(CLPA なし)の場合、PLPA の境界が SQA/ESQA の境界をコントロール。
 - コールドスタートのときは、初期 SQA/ESQA を展開する。
14. IEASVCxx よりユーザー-SVC テーブルを作る。
15. CLPA が指定されていたら、PLPA を作る。

- LPALSTxx データセット
 - SYS1.NUCLEUS から UIM で指定されたデバイスサポート。
16. FLPA と MLPA を作り、FLPA はページ固定し、どちらも必要に応じページプロテクトする。
 17. TYPE3,4SVC テーブルエントリーを完成。
 18. CSA=パラメータの処理。
 19. SRM(System Resource Manager)の初期化。
 20. タスク終了とSRB パージのための RTM を開始。
 - ここからマスタースケジューラーによって限られた機能のアドレス空間を作ることができるようになる。
 21. クロスメモリーサービスの初期化と、PCAUTH アドレス空間を作る。

Page28 NIP RIM 処理(続き)

22. RSM データスペースの初期化、RASP を作る。
23. システムトレースサービスの初期化、TRACE を作る。
24. タイミングサービスの初期化、必要に応じて TOD 時計のセット。
25. SVC ダンプサービスの初期化、DUMPSRV アドレス空間を作る。
26. XCF/XES サービスの初期化、XCFAS アドレス空間を作る。
27. GRS の初期化、GRS アドレス空間を作る。
28. SMS と PDSE サービスの初期化、SMXC と SYSBMAS アドレス空間を作る。
29. LNKLST をオープン -- SYS1.NUCLEUS をドロップ。
30. コンソールサービスの初期化、CONSOLE アドレス空間を作る。
 - フルファンクションのコンソールはまだ使用可能とならない。
31. WLM サービスの初期化、WLM アドレス空間を作る。
32. データ管理を初期化。
33. コンカレントコピーを初期化、ANTMAIN と ANTAS000 アドレス空間を作る。

34. UNIX システムサービスの初期化、OMVS アドレス空間を作る。

Page29 NIP RIM 処理(続き)

35. マスターカタログのクローズ。

36. カタログサービスの初期化、CATALOG アドレス空間を作る。

- MSI が完了するまでは、限定的な機能のみ。

37. NIP 終了。

- IPL パラメータエリア(IPA)を作る。
- NIP で使用し、もはや必要でないコントロールブロックの解放。
- 予期せぬエラーのためのトラップを解除。RTM 回復/再試行の全機能を使用可能とする。
- マスタースケジューラー処理へ LINK。

注:これらは NIP RIM の概略である。IPCS で見ることのできる、IPL 統計報告書に出る NIP RIM の完全なリストは、このレポートの終わりに掲載した。

Page30 マスタースケジューラー初期化

🚩 マスタースケジューラー初期化(MSI)の概略

- システム機能の初期化を完了。
- プライマリーサブシステム(JES2/JES3)との最終的な調整。

🚩 基本処理

- マスタートレースの初期化。
- フルファンクションコンソールを使用可能にする。
 - すべての MCS コンソールがここで使用可能となる。
- Sysplex レベルの ENF サービスの初期化、IEFSCHAS アドレス空間を作る。
- MSTR サブシステムの初期化。

- 共通 JES サービスの初期化、JESXCF アドレス空間を作成。
- アロケーションサービスの初期化、ALLOCAS アドレス空間を作成。
- マスターJCL をスタートするために、イニシエーターをアタッチ。

Page31 マスタースケジューラー初期化の詳細

1. MIH サービスの初期化。
2. ASM 初期化完了。
3. IOS ダイナミックパスの初期化、IOSAS アドレス空間の作成。
4. マスターのセキュリティ環境の初期化。
5. コンソール属性の初期化 (DEL=RD など)。
6. APPC サービスの初期化。
7. TSO サービスの初期化。
8. LOGREC ログストリーム・レコーディングの初期化。
9. ENF (Event Notification Facility) サービスの使用開始。
10. システムロガーサービスの初期化、IXGLOG アドレス空間の作成。
11. すべて使用可能な CP をオンライン。
 - マルチプロセッシングの開始。
12. SMF サービスの初期化、SMF アドレス空間の作成。

Page32 マスタースケジューラー初期化の詳細(続き)

13. PARMLIB 内の IEACMD00 と COMMNDxx のコマンドの発行。
 - CONSOLE アドレス空間で処理できるコマンドだけが今、実行される。
14. RTM サービスの初期化。

- LOGREC レコーディング。
 - アドレス空間の終了処理。
 - SVC ダンプの処理。
15. システムセキュリティ処理の初期化。
 16. 定義されたサブシステムの立ち上げ。
 - 初期化ルーチンを呼び出す。
 - 必要ならば、プライマリーJES サブシステムのために START を出す。
 17. プライマリーJES の STC と TSO の処理をホールドする。
 18. MSI が完了したことを表示。
 19. マスター・コマンド処理の初期化。
 - マスター内で実行がペンディングになっていた処理を行う。
 - マスターにより START コマンドは実行される。

Page33 マスタースケジューラー初期化の詳細(続き)

フルファンクションの機能をもったアドレス空間を作る。JES および MSTR サブシステムの下でスタートされた他のタスクは実行が開始する。

20. コマンド処理が可能になったメッセージを出す。
21. ペンディングになっていたアドレス空間の作成 (Master によって行われていなかった) を完了する。
 - フルファンクションの CATALOG
 - 元の CATALOG 空間 (限定的な機能のもの) は終了。
 - アドレス空間の機能を限定的なものから、フルファンクションへスイッチ。
22. プライマリーサービスが使用可能になったと、JES が知らせるのを待つ。
 - プライマリーJES が STC と TSO 処理をリリース。
 - システムログの Syslog/OPERLOG がスタート。

これですべての IPL 処理は完了。次の最後のステップは、ジョブエントリーサブシステム (JES2 または JES3) の起動と初期設定。