Vol.2010-ARC-189 No.14
Vol.2010-OS-114 No.14
2010/4/22

Tender の世代管理機能の設計

長井健悟[†1]　山本悠太[†2]
山内利宏[†2]　谷口秀夫[†2]

計算機の主記憶は揮発性であるため，計算機の電源切断により，主記憶上のデータは消失する．このため，我々は，***Tender*** オペレーティングシステムにおいて，計算機状態を永続化する動作継続制御を提案した．動作継続制御は，仮想記憶空間を利用して主記憶上のデータを不揮発性記憶媒体へ永続化するプレート機能を利用し，計算機停止前に行っていた処理の継続に必要なデータを永続化する．しかし，プレート機能により永続化可能なデータの状態は 1 つだけである．そこで，ここでは，複数の計算機状態を保存可能にする世代管理機能について述べる．世代管理機能は，計算機状態単位で全てのプレートの永続データの複製を複数世代分保存，復元可能にする．

Design of the Generation Management Function on *Tender* Operating System

Kengo Nagai,[†1] Yuta Yamamoto,[†2]
Toshihiro Yamauchi[†2] and Hideo Taniguchi[†2]

Data on main memory is no longer available when computer is turned off, because main memory is volatile. Therefore, we proposed persistent mechanism for computer proccessing on ***Tender*** operating system. Persistent mechanism uses ”plate” function to make persistent data which needed to continue processing before the shutdown. Plate manages a persistent data on virtual memory space. However, persistent mechanism can maintain only one state of persisted computer proccessing. Thus, we explain generation management function which enable to maintain two or more states of the computer proccessing. It permits plural generations of all persisted palte data to be maintained in each state of the compute proccessing.

1. はじめに

既存のオペレーティングシステム（以降，OS と略す）は，不揮発性記憶媒体である外部記憶装置上にファイルシステムを構築し，揮発性メモリ上のプログラムやデータをファイルとして保存することにより，プログラムやデータの再利用を可能にしている．また，既存 OS は，ファイル操作に関するインタフェースを応用プログラム（以降，AP と略す）に提供しており，AP は，各自で必要なデータを永続化する必要がある．このため，停電や電源ケーブルの接触不良などにより，計算機への電力供給が停止した場合，計算機が異常終了し，個々の AP は，データを永続化し損ねる可能性がある．また，既存 OS は，AP が利用する一部のデータのみを永続化し，AP の状態は永続化しない．このため，計算機を再起動しても AP の処理途中状態は復元されない．

計算機処理の中断と再開を可能にする機能として，Windows の休止機能や Linux の swsusp，TuxOnIce[1] のようなハイバネーション機能がある．ハイバネーション機能は，主記憶上の全てのデータを外部記憶装置に書き出し，計算機再起動後に保存したデータを主記憶に読み込む．これにより，計算機処理中断時の状態から処理を再開できる．しかし，外部記憶装置上の保存データの読み書き処理にかかる時間は，主記憶の大きさに比例する．また，ハイバネーション機能は，利用者が実行契機を与える必要があるため，予見できる停止には有効であるが，予見できない緊急停止には対応できない．

Tender オペレーティングシステム[2](以降，***Tender*** と略す) は，仮想記憶機構を利用した主記憶上のデータを永続化するプレート機能[3] を持ち，プレート機能を用いて主記憶上の計算機の処理状態 (以降，計算機状態と呼ぶ) を書き出し，復元できる．しかし，プレート機能により管理される永続データは，1 つのプレートに対して 1 つであるため，動作継続制御[4] において，最新の計算機状態のみが復元可能である．このため，ウィルスやワームなどの有害なプログラムの感染や AP の重大なバグの発生後，プレート書き出しを行った場合，異常のある計算機状態のみが保存される．このとき，プレートの復元により計算機状態を復元しても，正常な計算機状態を復元できない．

そこで，本論文では，永続データの世代管理機能を提案する．世代管理機能は，1 つのプ

†1 岡山大学工学部
Faculty of Engineering, Okayama University
†2 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University

Vol.2010-ARC-189 No.14
Vol.2010-OS-114 No.14
2010/4/22

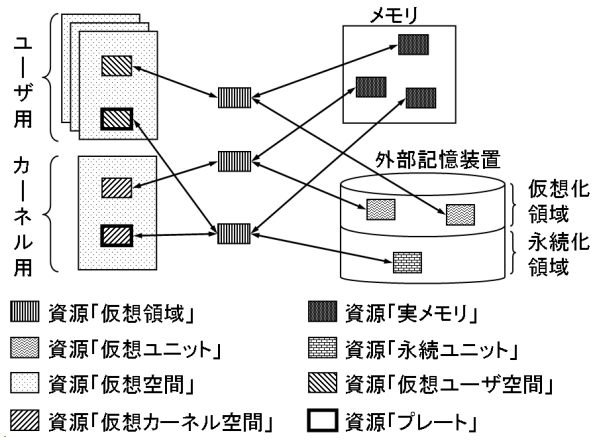


図 1 メモリ関連の資源の関係

レートに対応する永続データを複数保存可能にする．これにより，複数時点の計算機状態の保存を実現する．

ファイルの世代管理に関する既存技術として，Mac OS X の Time Machine[5] や rsnapshot[6] がある．世代管理機能はこれらの技術と同様に，ハードリンクを用いて，ファイルの世代管理を行う．ただし，世代管理機能は世代復元にハードリンクを用いることで，復元を複写レスで行う．これに対し，Time Machine や rsnapshot はバックアップファイルの容量を削減するためにハードリンクを用いるものの，復元には複写を必要とする．

2. *Tender* オペレーティングシステム

2.1 資源の分離と独立化

Tender では，OS が制御し管理する対象を資源として細分化し，資源の分離と独立化を行っている．資源の種類ごとに，資源を管理する管理表と，資源を操作するプログラムが存在する．資源は資源名と資源識別子によって識別される．

2.2 メモリ関連資源

Tender におけるメモリ関連資源の関係について，図 1 に示し，以下で説明する．

資源 「仮想領域」は，メモリ資源をイメージ化した資源である．メモリ上のデータは，資源「実メモリ」上，もしくは外部記憶装置上の領域に存在する．資源「実メモリ」は主記憶上の領域を表す資源である．資源「仮想ユニット」は，仮想化領域の管理単位である．仮想化領域とは，主記憶上のデータを一時的に保存するための領域であり，既存 OS のスワップ領域を利用して実現している．資源「永続ユニット」は，永続化領域の管理単位である．永続化領域とは，仮想記憶空間上のデータを永続化するために利用する領域であり，既存 OS のファイルシステム領域を利用して実現している．資源「仮想空間」は，特定のアドレス領域を持つ仮想的な空間である．仮想アドレスから実アドレスへのアドレス変換表に相当する．資源「仮想ユーザ空間」は，ユーザ用の仮想空間に存在するデータを表す．資源「仮想ユーザ空間」は，OS や AP が資源「仮想領域」で管理されるデータにアクセスするために，資源「仮想領域」をユーザ用の資源「仮想空間」に貼り付ける際に生成される．「貼り付ける」とは，仮想アドレスを実アドレスに対応付けることである．一方，仮想アドレスと実アドレスの対応付け解除を「剥がし」と呼ぶ．資源「仮想カーネル空間」は，カーネル用の仮想空間に存在するデータを表し，資源「仮想領域」をカーネル用の資源「仮想空間」に貼り付ける際に生成される．

2.3 プレート機能

Tender では，永続化の対象となる領域を「プレート」と呼ぶ．プレート機能とは，OS が自動的に仮想記憶空間上に存在する上のデータを外部記憶装置上の永続化領域へ書き出すことにより，データを永続化する機能である．***Tender*** では，プレート機能を資源「プレート」として実現している．プレート機能は以下の機能を持つ．

(機能 1) プレート書き出し機能

プレートを主記憶上から外部記憶装置上の永続化領域にファイルとして書き出すことにより，データを永続化する．

(機能 2) プレート復元機能

外部記憶装置上の永続化領域に存在するファイルを主記憶上に読み込むことにより，データを復元する．

AP は，必要に応じてプレートを自分の仮想記憶空間にマッピングして利用する．また，AP がプレート上のデータの書き出しを OS に依頼することもできる．

プレート機能は，計算機の電源投入時にプレートを復元する．これにより，プレートは，計算機の電源再投入後でも，計算機の電源切断前と同じ仮想記憶空間のアドレス領域に存在し続ける性質を持つ．

2.4 動作継続制御

動作継続制御は，計算機停止前の処理を計算機再起動後に継続して実行する機能を持つ．

Vol.2010-ARC-189 No.14
Vol.2010-OS-114 No.14
2010/4/22

動作継続制御の処理の流れを以下に示す.

(1) 仮想記憶空間上のデータの不揮発性化

プレート機能により，OS や AP が利用する仮想記憶空間上のデータをプレート化することで，永続化する.

(2) プレートの書き出し

プレート化された仮想記憶空間上のデータを外部記憶装置上の領域に書き出す.

(3) プレートの復元

OS の開始処理において，プレート管理部初期化時，外部記憶装置上にプレート管理表が存在している場合，プレート管理表に格納されている情報をもとに，外部記憶装置に保存されたデータから仮想記憶空間上にプレートを復元する．その後，復元したプロセスの動作継続を行う．具体的には，プレートの書き出しを依頼した AP の処理に切り替える.

3. 世代管理機能の基本方式

3.1 世代管理機能への要件と要望

世代管理機能とは，計算機状態を外部記憶装置上に複数保存する機能である．世代管理機能は以下の要件を満たす必要がある.

(要件 1) プレートの永続データを計算機状態単位で複数世代管理できること

複数の計算機状態を保存するためには，プレートの永続データを計算機状態単位で複数世代管理できる必要がある.

また，世代管理機能は，以下の要望を満たすのが望ましい.

(要望 1)　既存の資源管理部や AP への変更を最小限にとどめる

既存の資源管理部や AP の仕様に変更を加えた場合，バグの混入の原因になる．このため，プレートの永続化情報の世代管理を行うための機能を世代管理を行うプログラム構造に集約させ，既存の資源管理部や AP への変更を最小限にとどめる.

(要望 2) 永続データの保存と復元に要する処理時間を削減する

永続データの保存と復元に要する時間が長くなると，他の処理の実行を妨げ，利便性を低下させる．そこで，永続データの保存と復元に要する時間を削減し，利便性を向上させる.

3.2 世代管理機能の構成

世代管理機能は，以下の 3 つの機能によって構成される.

(機能 1) 世代システム

世代システムとは，ファイルシステムを利用し，全てのプレートの永続データの複製を複数世代分管理する機能である．具体的には，外部記憶装置上に永続化領域とは別にプレートの永続データを保存するための領域 (以降，世代保存領域と名づける) を確保し，各永続ユニットごとに対応付けられるファイルを複製する．また，世代管理機能は，永続データの世代情報を管理する必要がある．そこで，世代情報は，世代管理表に記録する．永続データの保存や永続データの削除によって世代管理表が更新された際，世代管理機能は，世代管理表を外部記憶装置上に書き出し，永続化する．ここで，世代管理表は，最新の状態のみ保存する必要があり，過去の状態を保存する必要は無い．このため，世代管理表は，他のプレートと同じ永続化領域に保存せず，世代保存領域への複写も行わない．これにより，3.1 節の (要件 1) を満たす．また，永続化領域上の永続データの保存と復元は，プレート管理や永続ユニット管理から独立して行われる．したがって，他の資源管理への変更を行う必要は無く，(要望 1) を満たす.

(機能 2) 世代保存機能

世代保存機能の処理の流れを図 2 に示す．世代保存機能とは，現在の計算機状態を 1 つの世代として保存する機能であり，計算機状態を主記憶上から永続化領域上に保存し (図 2(1))，これを新たに確保した世代保存領域に複写する (図 2(2)).

(機能 3) 世代復元機能

世代復元機能の処理の流れを図 3 に示す．世代復元機能とは，指定した世代の計算機状態を復元する機能であり，世代保存領域上に保存してある計算機状態を永続化領域に複写し (図 3(1))，プレート復元を行うことにより主記憶上に読み込む (図 3(2)).

以上の機能を実現するために，ファイルシステムのディレクトリ構成，世代保存領域の実現方式，およびファイル複写方式について検討する必要がある．また，(要望 2) を満たすため，ファイル複写を高速化する手法について検討する必要がある.

3.3 世代システム

3.3.1 ディレクトリ構成

Tender が利用するファイルシステムのディレクトリ構成を図 4 に示す．永続化領域，世代保存領域，および世代管理表は，ファイルシステム上の/tender/plate 以下の領域に存在するものとする．プレートの永続データを保存する永続化領域として，/tender/plate/current 以下の領域を用いる．世代保存領域として，/tender/plate/XXX 以下の領域を用いる．ここで，XXX は，世代名とする．世代管理表は，/tender/plate 直下に保存する.

3.3.2 世代保存領域の実現方式

プレートの状態を保存したファイルを世代保存領域に保存する．世代保存領域の実現方式

Vol.2010-ARC-189 No.14
Vol.2010-OS-114 No.14
2010/4/22

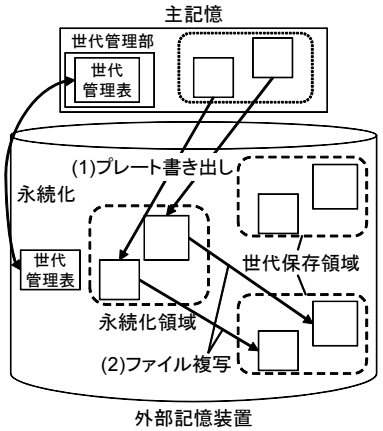


図 2 世代保存機能の処理の流れ

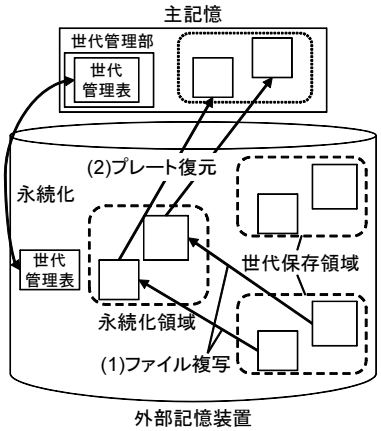


図 3 世代復元機能の処理の流れ

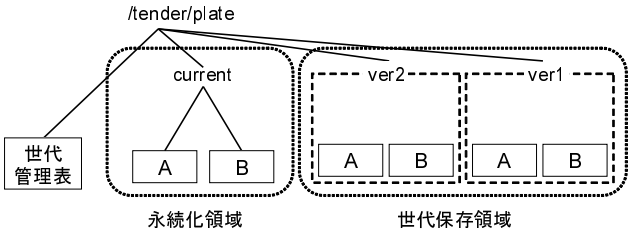


図 4 世代システムのディレクトリ構成

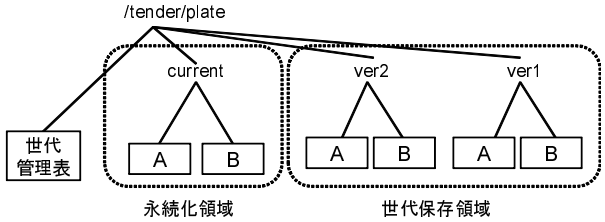
として，以下の 2 つが考えられる．

(方式 1) ディレクトリ方式

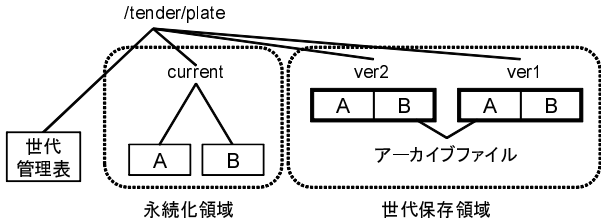
ディレクトリ方式の概要図を図 5(a) に示す．ディレクトリ方式では，世代保存領域用のディレクトリを生成し，プレートの永続データを保存したファイルをそのディレクトリに複写する．復元時には，世代保存領域用のディレクトリから永続化領域上へプレートの永続データを保存したファイルを複写する．

(方式 2) アーカイブ・圧縮方式

アーカイブ・圧縮方式の概要図を図 5(b) に示す．アーカイブ・圧縮方式では，プレート



(a) ディレクトリ方式



(b) アーカイブ・圧縮方式

図 5 世代保存領域の実現方式

の状態を保存した複数のファイルを 1 つのファイルにアーカイブ，または圧縮して保存し，世代保存領域として扱う．

保存領域確保方式の比較を以下に示す．

(方式 1) は，プレートの永続データをそれぞれ単独のファイルとして保存するため，データの追加や移動が容易である．よって，世代保存時にコピーオンライトや差分保存を用いることが容易となる．これらを用いることでファイル複写データ量の削減が期待できる．

一方，(方式 2) は，全プレートの永続データを単一のファイルとして保存するため，世代保存処理において，アーカイブ・圧縮処理が必要となり，処理のオーバヘッドが発生する．また，データの追加や移動は困難である．さらに，コピーオンライトや差分保存を用いることは難しい．

以上のことから，コピーオンライトや差分保存による複写データ量の削減と世代保存と世代復元に要する処理時間の削減が行える (方式 1) を採用する．

3.3.3 ファイル複写方式

3.2 節で述べたように，世代保存処理と世代復元処理では，永続化領域と世代保存領域の

Vol.2010-ARC-189 No.14
Vol.2010-OS-114 No.14
2010/4/22

間でファイル複写が行われる．世代保存処理や世代復元処理の際，毎回全てのファイルを複写すると世代保存と世代復元に要する処理時間は膨大になる．そこで，ファイル複写データ量を削減し，世代保存と世代復元に要する処理時間を短縮する．ファイル複写データ量を削減する手法として，以下の方法が考えられる．

(方式 1) コピーオンライトでのファイル複写

世代保存処理や世代復元処理時にファイル複写を行わず，ファイルの更新を行う際にファイル複写を行う．これにより，ファイル複写データ量の削減が期待でき，世代保存と世代復元に要する処理時間を短縮できる．また，世代保存処理における永続化領域から世代保存領域へのファイル複写だけでなく，世代復元処理における世代保存領域から永続化領域へのファイル複写にも適用可能である．ただし，ファイル更新処理時のファイル複写を行うため，ファイル複写を行うか否かの判定を行う必要があり，ファイル更新処理時間は長くなる．

(方式 2) 前回の世代保存からの差分のみファイル複写

世代保存処理の際，全てのファイルを複写せず，前回の世代保存からの差分のみ保存することで，世代保存と世代復元に要する処理時間を短縮する．ただし，初回の世代保存処理と世代復元処理は，全ての永続データを複写する必要があり，複写データ量を削減できない．

以上のことから，世代保存処理と世代復元処理の両方に要する処理時間を短縮可能な (方式 1) を採用する．

3.3.4 コピーオンライトの実現方式

コピーオンライトを実現するには，世代保存からファイル複写までの間，世代保存された永続データが永続化領域上の永続データとして参照可能である必要がある．そこで，世代保存された永続データを永続データとして参照可能にするために，NTFS や FFS など，多くのファイルシステムで提供しているハードリンクを利用する．具体的には，図 6 に示す通り，プレートの永続データを保存したファイルのハードリンクを永続化領域用ディレクトリと世代保存領域用ディレクトリの両方に生成し，永続化領域と世代保存領域間でファイルを共有する．永続化領域上の永続データを削除する際や世代削除により世代保存領域上の永続データを削除する場合，永続化領域用ディレクトリ，もしくは世代保存領域用ディレクトリに存在するプレートの永続データを保存したファイルのハードリンクを削除する．これにより，他の世代保存領域とファイルを共有している場合でも，ファイルは削除されない．

ファイルを更新する場合において，前回の世代保存，または前回の世代復元からはじめて更新される永続データに対し，ファイル複写を行う必要がある．そこで，世代保存からはじめて更新される永続データか否かを判定するため，ファイルのリンク数を用いる．具体的に

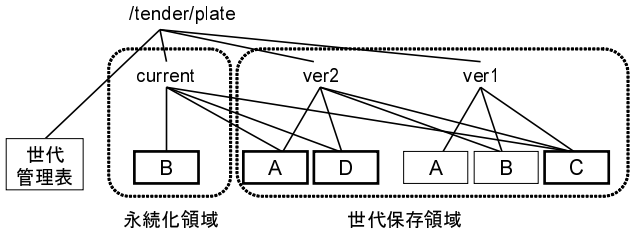


図 6 ハードリンクによるプレートの状態の保存

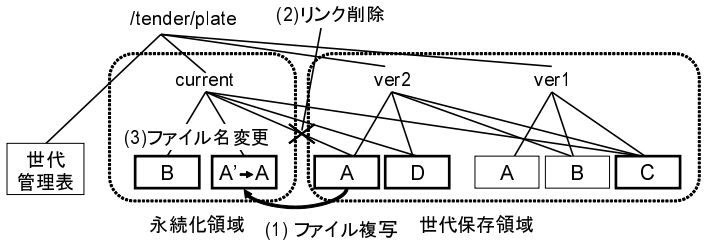


図 7 ファイル複写の流れ

は，ファイル更新の際，ファイルのリンク数を確認し，リンク数が 2 以上のファイルを更新する場合，ファイル複写を行った後にファイル更新処理を行う．ファイル複写の流れを図 7 に示し，以下で述べる．

(1) 一時ファイルを生成し，更新対象ファイルの内容をを一時ファイルに複写する．
(2) 更新対象ファイルのハードリンクを削除する．
(3) 複写したファイルのファイル名を更新対象ファイルのファイル名に変更する．

4. 世代管理機能の実現方式

4.1 世代管理機能の実現における要件

世代保存において，永続化領域に計算機状態を書き出す際，単にプレート機能を用いて全てのデータを永続化するだけでは，世代復元により，計算機処理を正常に継続させることはできない．全プレート書き出しにより，永続化したデータを用いて計算機処理を正常に継続するための要件を以下に示す．

(要件 1) データ間の整合性を保証すること

Vol.2010-ARC-189 No.14
Vol.2010-OS-114 No.14
2010/4/22

依存関係のある複数のデータ間で不整合がある場合，そのデータを利用する AP は正常に動作しない．このため，復元した永続データを用いて正常に計算機処理を継続するためには，永続化したデータにおいて，データ間の整合性を保証する必要がある．

(要件 2) 実入出力終了待ちのプロセスが存在しないこと

実入出力終了待ちのプロセスは，デバイスからの割り込みにより覚醒される．しかし，実入出力終了待ちのプロセスが存在する状態を保存しても，復元後にデバイスからの割り込みは発生しない．このため，永続データ復元後は，実入出力終了待ちのプロセスは覚醒されないため，処理を継続することはできない．したがって，実入出力終了待ちのプロセスが存在しない状態で全プレート書き出しを行う必要がある．

(要件 3) ファイルシステムのメタデータを操作中でないこと

プロセスがプレートの生成や削除などの操作を OS に依頼した場合，OS 処理中にファイルの生成や削除により，ファイルシステムのメタデータを操作する．メタデータ操作中に計算機状態を書き出す場合，計算機状態復元後にファイル操作を正常に継続するためには，ファイルシステム全体を整合性を保った状態で保存復元できる必要がある．プレート機能は，単に仮想記憶空間上のデータを永続化するための機能であり，ファイルシステムの状態を保存することはできない．したがって，OS がファイルシステムのメタデータを操作していない状態で全プレート書き出しを行う必要がある．

上記の要件を満たさなければ，計算機処理を正常に継続できないため，単にプレート機能を用いてデータを永続化するだけでは，世代復元により，計算機処理を再開できない．

4.2 復元した計算機処理を継続するための対処

各要件に対する対処について以下に示す．

4.2.1 データ間の整合性を保証することへの対処

データ間に不整合が生じる原因として，以下の要因がある．

(要因 1) プロセスの処理実行

プロセスの処理実行により，プロセスが利用するデータの更新が行われる可能性がある．依存関係のあるデータが別々に保存された場合，データ間に不整合が生じる．

(要因 2) 割り込みルーチンの実行

割り込みルーチンの実行により，OS が管理するデータの更新が行われる可能性がある．例えば，タイマ割り込みにより，スケジューリングのような周期的に実行されるルーチンが呼び出される可能性がある．

全プレート書き出し中にこれらの要因が発生しなくなれば，データ間に不整合が生じることはなくなる．そこで，全プレート書き出し時，プロセッサを占有し全プレート書き出し以外の処理が実行されないようにする．具体的には，以下の対処を行う．

(対処 1-A) プリエンプションを禁止する．

スケジューラにプリエンプションを禁止するように依頼する．これにより，プリエンプションによる他プロセスへの切り替えを防ぐ．

(対処 1-B) 割り込みを禁止する．

割り込みマスクを設定し，割り込みを禁止する．これにより，割り込みルーチンは実行されなくなる．

(対処 1-C) デバイスへの実入出力をポーリング方式で行う．

全プレート書き出し時，外部記憶装置上へデータを書き出す．このとき，ビジーウェイトにより入出力処理完了を待つポーリング方式で行う．これにより，実入出力処理開始後に，プロセスを休眠させる必要は無くなり，他プロセスへ切り替わらない．

4.2.2 実入出力終了待ちのプロセスが存在しないことへの対処

全プレート書き出し時に，実入出力終了待ちのプロセスが存在しないようにするには，入出力デバイスをプレート書き出し以外の処理で利用されないようにすればよい．そこで，全プレート書き出し時，入出力デバイスを占有し，他のプロセスの処理による実入出力が実行されないようにする．具体的には，以下の対処を行う．

(対処 2-A) 現在実行中の実入出力処理の完了を待つ．

現在実行中の実入出力処理が完了するまで全プレート書き出しを中断する．実入出力処理中でなくなってから全プレート書き出しを開始する．

(対処 2-B) 全プレート書き出し処理以外の処理による実入出力処理の実行を中断する．

新たにデバイスへの実入出力処理を実行する場合，全プレート書き出しが終了するまで処理を中断する．

4.2.3 ファイルシステムのメタデータを操作中でないことへの対処

全プレート書き出し時，ファイルシステムの機能の利用を制限し，他のプロセスの処理により，ファイルを操作されないようにする．具体的には，以下の対処を行う．

(対処 3-A) 現在実行中のファイル操作の完了を待つ．

現在実行中の実入出力処理が完了するまで全プレート書き出しを中断し，ファイルシステムのメタデータ操作中でなくなってから全プレート書き出しを開始する．

(対処 3-B) 全プレート書き出し処理以外の処理によるファイル操作を中断する．

新たにファイル操作を行う場合，全プレート書き出しが終了するまで処理を中断する．

Vol.2010-ARC-189 No.14
Vol.2010-OS-114 No.14
2010/4/22

4.2.4 実現における課題

4.2.1 項，4.2.2 項，および 4.2.3 項より，全プレート書き出し時，プロセッサや入出力デバイスなどのハードウェア資源を占有し，ファイル操作実行を制限する必要がある．このためには，ハードウェア資源の占有制御を行う機構が必要になる．4.3 節と 4.4 節で，実現したハードウェア資源の占有制御機構について述べる．

4.3 ハードウェア資源の占有制御機構

4.3.1 機　　能

ハードウェア資源の占有制御機構に必要な機能を以下に示す．

(機能 1) ハードウェア資源の占有機能

プリエンプションの禁止，割り込みの禁止, およびポーリング方式へのディスク I/O 方式の変更を行い，ハードウェア資源を全プレート書き出しを行うプロセス以外のプロセスや割り込みルーチンにより利用されないようにする．

(機能 2) ハードウェア資源の占有解除機能

プリエンプションの禁止解除，割り込み方式へのディスク I/O 方式の変更，および割り込みの禁止解除を行い，ハードウェア資源を全プレート書き出しを行うプロセス以外のプロセスや割り込みルーチンにより利用されるようにする．

(機能 3) ファイル操作や実入出力の実行管理機能

ファイル操作や実入出力処理の実行状況を監視する必要がある．そこで，ファイル操作や実入出力処理の実行を制御するため，ハードウェア資源の占有制御機構がファイル操作や実入出力処理を実行するためのインタフェースを提供し，ファイル操作や実入出力処理を行う際は，必ずそのインタフェースを呼び出すことにする．

具体的には，以下の処理を行う．

(1) 実行中のファイル操作数と実入出力処理数の管理

実行中のファイル操作数と実入出力処理数をカウントし，ファイル操作や実入出力処理が実行中か否かを管理する．

(2) ファイル操作と実入出力処理の中断

ハードウェア資源の占有前に現在実行中のファイル操作や実入出力処理が完了するまで，全プレート書き出しを延期する．ここで，ファイル操作と実入出力処理の完了待ち中に新たにファイル操作と実入出力処理が発生した場合，新たに発生したファイル操作と実入出力処理を中断する．

(3) ファイル操作と実入出力処理の再開

全プレート書き出し前にファイル操作や実入出力処理を中断した場合，ハードウェア資源の占有解除機能において，ファイル操作や実入出力処理を再開する．

4.3.2 課題と対処

(課題 1) ディスク I/O 方式の指定方法

ディスク I/O をポーリング方式で行うためには，ディスク I/O を行う際，デバイスドライバにディスク I/O 方式を通知する必要がある．

そこで，ハードウェア資源の占有制御機構は，ファイル操作や実入出力処理を実行するためのインタフェースを提供する．そこで，UNIX ファイル制御とディスクドライバのインタフェースの引数でディスク I/O 方式を指定できるようにする．ハードウェア資源の占有制御機構がファイル操作を行う際，インタフェースの引数でディスク I/O 方式を指定可能になる．

(課題 2) プレート復元後の処理

プレート復元により，計算機状態は，全プレート書き出し時の状態に復元される．全プレート書き出し時はハードウェア資源を占有しているため，プリエンプションが禁止になっている．また，全プレート書き出し前にファイル操作や実入出力処理を中断している可能性もある．

そこで，プレート復元後に，ハードウェア資源の占有解除を行い，プリエンプションの禁止を解除し，ファイル操作や実入出力処理を再開する必要がある．

4.4 資源「システム」

Tender において，プログラムを実行する際に利用するハードウェア資源の構成を表す資源として資源「システム」を実現する．資源「システム」は，プロセッサ，外部記憶装置，通信用デバイスなどのハードウェア資源で構成される．資源「システム」は，システム管理処理部により管理される．ただし，本論文では，CPU と外部記憶装置から構成されるシステムを 1 つのみ作成でき，Myrinet や Ethernet 用の通信用デバイスは管理対象外とする．また，システムが管理するハードウェア資源の設定方法や複数のシステムの管理方法については今後の課題とする．

システム管理処理部は，資源「システム」を構成するハードウェア資源の利用を制御する機能を持つ．システム管理と他の資源管理処理部との関係図を図 8 に示す．具体的には，2 つの機能を提供する．

(機能 1) ハードウェア資源への入出力処理機能

システム管理処理部は，資源「システム」を構成するハードウェア資源への入出力処理機

Vol.2010-ARC-189 No.14
Vol.2010-OS-114 No.14
2010/4/22

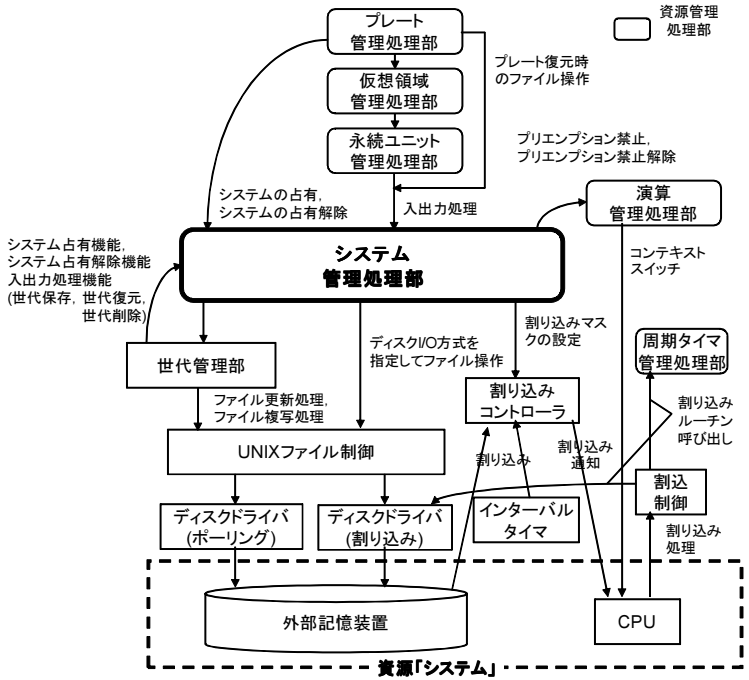


図 8 システム管理と他の資源管理との関係

能を提供する．各資源管理処理部は，この機能を利用して入出力処理を行う．入出力処理機能の一つとして，ファイル操作を提供する．図 8 の例において，永続ユニット管理処理部は，プレートの内容の読み込みや書き出しを行う際，ファイル操作を行うためにシステム管理処理部の機能を利用する．

(機能 2) ハードウェア資源の占有制御機能

システム管理処理部は，資源「システム」を構成するハードウェア資源の占有制御機能を提供する．システムの占有を要求された場合，システム管理処理部は，システム占有解除まで他のプロセスやデバイスの割り込みルーチンにより，システムが管理するハードウェア資源を利用されないように制御する．プロセッサへのプロセスの割り当ては，演算管理処理部によって管理されるため，プロセッサを占有する際，演算管理処理部にプリエンプションの禁止を依頼する．また，システムが管理するハードウェア資源への入出力処理実行依頼があった場合，システム占有解除まで入出力処理を中断する．

5. おわりに

Tender の動作継続制御において，永続データを複数世代保存可能にする世代管理機能について述べた．世代管理機能は，全てのプレートの永続データの複製を複数世代分保存し，復元することを可能にする．この機能により，計算機状態単位でのプレートの永続データの保存が可能になる．また，永続データを指定して，計算機状態を復元し，計算機処理を再開できる．永続化領域と世代保存領域間のデータ複写は，ハードリンクを用いたコピーオンライトで実現した．これにより，世代保存処理と世代復元処理に要する処理時間を短縮できる．

全プレート書き出しにより，永続化データを正常に保存するためには，全プレート書き出し時にプロセッサやデバイスなどのハードウェア資源を占有する必要がある．この対処として，ハードウェア資源を資源「システム」により管理し，システム管理処理部がハードウェア資源の占有制御を行う手法について述べた．各資源管理処理部は，システム管理処理部に入出力処理を要求する．システム管理処理部は，ハードウェア資源を占有するために入出力処理を中断するか否かを判定する．これにより，資源「システム」を構成するハードウェア資源の占有を制御できる．

残された課題として，世代管理機能の性能，および，資源「システム」によるプロセッサと外部記憶装置以外のハードウェアの管理と占有方法の検討がある．

参 考 文 献

1) Cunningham, N.: TuxOnIce. http://www.tuxonice.net.
2) 谷口秀夫, 青木義則, 後藤真孝, 村上大介, 田端利宏：資源の独立化機構による ***Tender*** オペレーティングシステム，情報処理学会論文誌， Vol.41, No.12, pp.3363–3374 (2000).
3) 的野　司，田端利宏，谷口秀夫：***Tender*** におけるプレート機能を利用した資源の永続化機構の設計，情報処理学会研究報告，2003-OS-93, Vol.2003, No.42, pp.147–154 (2003).
4) 大本拓実，田端利宏，谷口秀夫：仮想記憶空間上のデータ不揮発性化による計算機処理永続化方式の提案，情報処理学会研究報告，2007-OS-104, Vol.2007, No.10, pp.25–32 (2007).
5) アップル：Mac OS X Snow Leopard. http://www.apple.com/jp/macosx/.
6) Rubel, M.: rsnapshot. http://rsnapshot.org/.