2007－OS－105 (2)
2007／4／5

AnT オペレーティングシステムにおける プロセス生成制御機構

仁科 匡人，野村 裕佑，田端 利宏，乃村 能成， 谷口 秀夫

岡山大学 大学院自然科学研究科

AnT オペレーティングシステムは，マイクロカーネル構造を基本構造としている．このため，ファイル管理やデバイスドライバといった OS 機能の一部を外コアと呼び，プロセスとして動作させている．これらの外コアを利用環境に合わせて起動することで，様々な利用環境に適応可能としている．ここでは，プロセス生成制御機構について述べる．本機構は，計算機の利用環境を把握し適切な外コアの起動，および効率的なプロセスの走行制御を行う．

Process creation mechanism for *AnT*

Tadato NISHINA, Yusuke NOMURA,
Toshihiro TABATA, Yoshinari NOMURA and Hideo TANIGUCHI

Graduate School of Natural Science and Technology, Okayama University

AnT is an operating system based on micro-kernel architecture. Filesystem, device driver and other operating system modules run as a process named external-core. ***AnT*** can adapt various kind of computer environment by changing set of external-core. In this paper, we present process creation mechanism for ***AnT*** to make ***AnT*** adapt computer environment and create process need by ***AnT***. And we present efficient mechanism of process control.

1. はじめに

マイクロプロセッサや入出力ハードウェアの進歩には目覚しいものがある．また，様々な場面で計算機が必要となり，提供するサービスの種別も飛躍的に増大している．

そこで，我々は，走行モード変更[1]やゼロコピー通信，適応制御[2]といった機能を持つ***AnT***オペレーティングシステム[3](以降，***AnT***と呼ぶ)を開発している．プログラム構造としてマイクロカーネル構造を採用し，デバイスドライバやファイルシステムといった機能をプロセス(外コア)として動作させる．外コアの起動，終了，および走行形態を制御することで，ハードウェアとサービスを様々に組み合わせたシステムについて，効率的な管理制御を目指している．

ここでは，***AnT***オペレーティングシステムにおけるプロセス生成制御機構について述べる．まず，***AnT***の基本構造について述べる．次に，プロセス生成制御機構に対する要求を明らかにし，プロセス生成制御機構の詳細について述べ，本機構により要求を満足したこ

とを述べる．

2. *AnT* の基本構造

AnTの基本構造を図1に示し，以降に説明する．

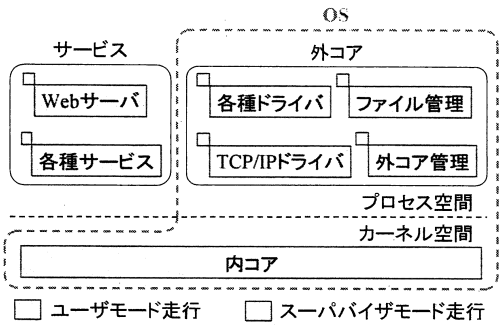

図 1　***AnT*** の基本構造

プログラムは，OSとサービスからなる．OSは，内コアとプロセスとして動作する外コアからなる．サービスは，利用者の要求するサービスを提供するプロセスからなる．各々のプログラムについて，以下に説明する．

(1)内コア

システムの最小限な機能の動作を保証するプログラム部分である．主な機能として，メモリ領域管理，プロセスの実行権制御部がある．

(2)外コア

適応したシステムに必須なプログラムであり，動的に再構成可能な構造を有する．主な機能として，デバイスドライバやファイル管理がある．

(3)サービス

利用者の要求する機能を提供する部分である．

3. 要求条件

プロセス生成制御機構への要求として以下のものがある．

(要求1)OS動作に必須なプロセスの生成

AnTでは，マイクロカーネル構造を採用しており，OSの動作に必須な機能には，プロセスとして動作するものがある．このため，OSの開始処理中にOS動作に必須なプロセスを生成することが要求される．

(要求2)システムに必要なプロセスの生成

それぞれのシステムで入出力ハードウェアの構成や必要なサービスの種類は異なっている．計算機の資源を効率的に利用するために，それぞれのシステムの環境を把握し，必要なプロセスを選んで生成する必要がある．

(要求3)デバイスドライバの起動制御

デバイスドライバは，プロセスとして動作している．このため，ユーザプログラムと同様にデバイスドライバを生成することが可能である．しかし，ユーザによって任意にデバイスドライバを生成された場合，OSはデバイスドライバを管理することができなくなる．このため，不用意なデバイスドライバの起動を抑止する必要がある．

(要求4)効率的なプロセスの走行

マイクロカーネル構造を持つ***AnT***では，OS機能の一部をプロセスとして動作させる．このため，モノリシックカーネル構造を持つOSと比べてプログラム間での通信が頻発しオーバヘッドが大きくなってしまう．そこで，プロセス走行におけるオーバヘッドを削減し，効率的なプロセスの走行が要求される．

4. プロセス生成制御機構

4.1. 基本機能

AnT におけるプロセス生成制御の器本機構を図 2 に示し，プロセス生成制御に関わる処理を表 1，管理表を表 2 に示す．

ここで，coreID とはデバイスドライバの種別を識別する情報である．また，ファイル管理は，デバイスを特殊ファイルとして AP に見せている．表 1 に示す各処理機能が連携し動作することでプロセス生成制御機構を実現する．***AnT*** におけるプロセスの生成の契機は，

(1) OS 立ち上げ時

(2) ハードウェアの検出時

(3) AP のデバイス利用要求時

(4) ユーザからの生成要求時

の 4 つに分類できる．それぞれの契機を以下に説明する．

AnT では，(1)OS の立ち上げ時に内コアが初期化プロセス（図 2 の init）を生成する．

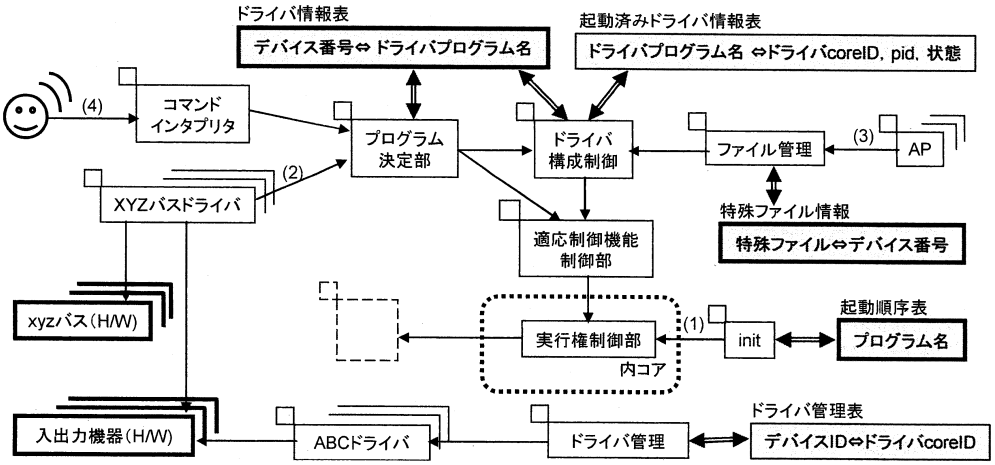

図 2　プロセス生成制御の基本機構

表 1　プロセス生成制御機構を構成する各処理機能

処理名	機能
コマンドインタプリタ	ユーザからの生成要求を受け付ける.
バスドライバ	ハードウェアを検出し，対応するドライバの生成要求を行う.
応用プログラム(AP)	デバイスを利用するプロセスであり，デバイスの利用要求を行う.
プログラム決定部	デバイス情報から対応するデバイスドライバを選択し生成要求を行う．また，サービス要求とH/W検出を統一的に扱いプログラム間の関連の深いものは同時にプロセス生成要求を行う.
ファイル管理	デバスの利用要求をドライバ構成制御へ通知する.
ドライバ構成制御	デバイスドライバの起動終了を制御し，デバスドライバの構成を制御する.
適応制御機能制御部	プロセスの動作履歴を収集し動作履歴に基づきプロセスの走行形態を制御する.
実行権制御部	プロセスの生成や終了を行う内コアの機能部分である.
ドライバ管理	ドライバの初期化や管理を行う.
初期化プロセス(init)	OS 動作に必須なプロセスを生成する.

表 2　プロセス生成制御機構の管理表

管理表名	内容
特殊ファイル情報	特殊ファイル名とデバイス番号の対応関係を保持している.
ドライバ情報表	デバイス番号とドライバプログラム名の対応関係を保持している.
起動済みドライバ情報表	起動しているドライバのプログラム名と coreID，pid，デバイスドライバの状態といったデバイスドライバの情報を保持している.
ドライバ管理表	デバイス番号と coreID の対応関係を保持している.
起動順表	初期化プロセスの起動するプロプログラムの名前と順序を保持している.

初期化プロセスは，生成されると，特定のプロセスを生成する．ここで生成される特定のプロセスについては，5.1 節で述べる.

AnTでは，OS立ち上げ時間の削減や占有メモリ量の削減のために，デバイスドライバの起動制御[4]を行う．このため，***AnT***ではOS立ち上げ時に全てのデバイスドライバプロセスを生成するのではなく，必要なときに必要な

デバイスドライバプロセスを生成する．デバイスドライバプロセスを生成する契機の1つとして(2)ハードウェアの検出がある．ハードウェアの検出は，OS開始時に既に接続されているPCIデバイスのようなハードウェアの検出とOS動作中に接続されるPnPデバイスのようなデバイスの検出に分けることができる．もう 1 つの契機として(3)APのデバイス利用要求がある．APがデバイス操作を行う場合，まずデバイスに対し利用要求を行う．この時，利用要求を行なったデバイスに対応するデバイスドライバプロセスが生成されていない場合には，利用要求を契機として対応するデバイスドライバプロセスを生成する．このようにシステムの環境を把握し，必要なドライバプロセスを生成することにより(要求 2)を満足することができる．

コマンドインタプリタを用いて(4)ユーザがプロセスの生成要求を行なったときに，要求されたプロセスを生成する．

それぞれの生成方法については 5 章で詳細に述べる．

4.2. 適応制御機能

4.2.1. 基本機構

適応制御機能は，占有メモリ量の削減やプロセスの効率的な動作を目的として，プロセスの動作情報を収集し，プロセスの制御に反映させる機能である．本機能は，処理内容により大きく 3 つの処理部に分類できる．また，各処理部が利用する管理表も大きく 3 つに分類できる．この様子を図 3 に示し，以下に説明する．

収集部は制御に必要な情報を逐次記録する．管理部は収集した情報の加工や更新，保存，復元を行う．制御部は収集した情報を基にプロセスの動作を制御する．これらの処理部において，情報の逐次記録は内コアの他の処理部と深く関係する．処理オーバヘッドの削減のため収集部のみ内コアに配置する．

状況表は収集した情報を時系列に記録した表である．履歴表は，状況表の情報を下に一定時間間隔ごとに整理した情報を持つ表である．制御表は制御に必要な観点から整理更新された情報を保持する．

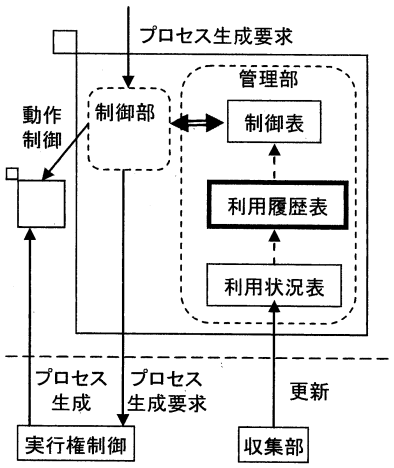

図 3　適応制御機能

これらの処理部の処理内容や表の情報は 4.2.2 項で例を示し説明を行う．

4.2.2. 動作例

プロセスの走行形態の制御として，走行モードの設定やメモリ常駐化の設定を自動的に行うことが考えられる．

走行モード変更機能は，プロセス空間プログラムの走行モードを変更し，変更されたプログラムと内コアの連携方式を変更する機能である．プロセス空間のプログラムをスーパバイザモードで動作させることにより，カーネルコールを関数呼び出しとして行なわせることでプロセスの高速な動作を可能とする．しかし，スーパバイザモードでの走行は OS と同レベルで走行するためプロセスが誤動作した場合，OS へ与える影響が大きくなってしまう．このため，スーパバイザモードで動作させるには高い信頼性が必要である．そこで，信頼性が高いものをスーパバイザモードで走行させるものとする．ここでは，簡単のため信頼性の高さを異常終了の回数で判断するものとする．

メモリの常駐化は，プロセスの利用しているメモリ領域をページアウト禁止にし，実メモリ上に常駐化させ，特定のプロセスの利用するメモリ領域のページアウトを抑止する．これにより，ページングのオーバヘッドを削

減し，プロセスを高速に動作させることが可能となる．しかし，実メモリは有限であり，全てのプロセスのメモリ領域を常駐化させることは難しい．そこで，利用度の高いプロセスのメモリを常駐化させるものとする．ここでは，簡単のためプロセスの利用度を一定時間内のプロセスの合計動作時間で表し，プロセスの合計動作時間が多いものを利用度が高いとする．

走行モード変更やメモリ常駐化の判断を行うために，プロセスの異常終了回数や合計動作時間といった情報が必要である．このため収集部により，プロセスの異常終了とプロセスの起動時刻，終了時刻の情報を取得し利用状況表に記録する．利用状況表に記録された情報を整理し，利用履歴表を更新する．利用履歴表は外部記憶装置に記録され情報を永続化して保持しておく．さらに，制御に必要な観点から情報を整理し，制御表を更新する．管理表の更新は，一定の間隔で周期的に行う．

適応制御機能がプロセスの生成要求を受け取ると，実行権制御部に対しプロセス生成システムコールを発行する．このとき，制御表を参照し，合計利用時間が閾値以上のプロセスはメモリ常駐化を行ない，異常終了回数が閾値以下のプロセスはスーパバイザモードで走行させる．

本機能により，プロセスの動作情報に基づきプロセスの走行形態を制御することで，プロセス動作時のオーバヘッドを削減し，プロセスを効率的に動作させることが可能となる．これにより，(要求 4)を満足させることができる．

5. プロセスの生成制御

5.1. OS 立ち上げ時のプロセス生成制御

AnTでは，OSの開始処理中に初期化プロセスを起動する．初期化プロセスは，起動するプログラムのリストを参照しOSの動作に最低限必要となるプログラムを起動する．このリストを初期起動リストと呼ぶ．これらの処理により，システムに必須なプロセスを生成し(要求1)を満足するとことができる．初期化プロセスが必須プロセスを生成した直後のシステムの構成と依存関係を図4に示す．図4の番号はプロセスの生成順序である．

初期化プロセスによるプロセス起動順序は，相互の連携を考慮して，ドライバ管理，ルートデバイスドライバ，ファイル管理，適応制御，ルートバスドライバ，ドライバ構成制御，プログラム決定，ルートコマンドインタプリタの順である．なお，ルートコマンドインタプリタは最初に起動するコマンドインタプリタとする．これらの初期化プロセスにより起動されたプロセスには特権を与える．

初期化プロセスにより起動されたルートバスドライバは，他のバスやハードウェアを検出し，バスドライバやデバイスドライバを起動する．起動の流れについては5.2.1項で詳細に述べる．

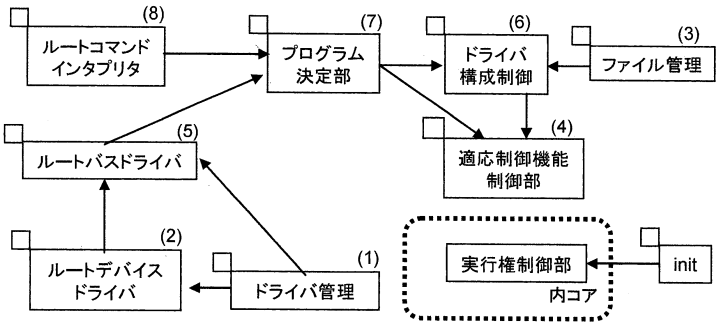

図 4　必須プロセス生成後のシステム構成

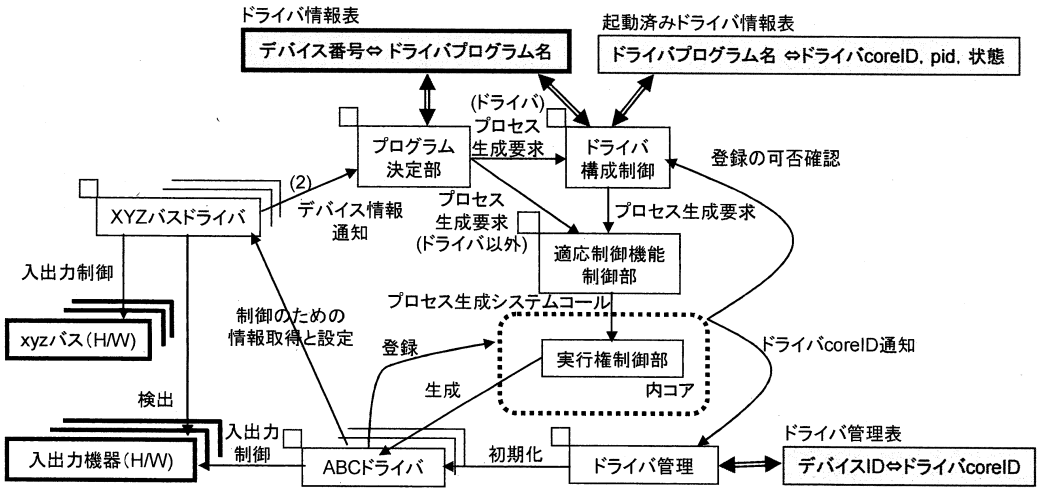

図 5　ハードウェアの検出を契機としたドライバプロセスの生成

5.2.　デバイスドライバプロセスの生成制御

5.2.1. ハードウェアの検出を契機とした生成

ここでは，ハードウェアの検出を契機として，対応するデバイスドライバプロセスを生成する流れについて説明する．ハードウェアの検出は，

(1) OS 立ち上げ時に検出

(2) OS 動作中に検出

のいずれかの場合に起こる．ハードウェア検出を契機としたプロセス生成の様子を図 5 に示し，プロセス生成の流れを以下に説明する．

(1) OS立ち上げ時に検出

初期化プロセスにより起動されたルートバスドライバはハードウェアを検出し，デバイス情報をプログラム決定部に通知する．プログラム決定部はドライバ情報表を参照し，ドライバプロセスや関連するサービスプロセスの生成要求を行う．ドライバ構成制御はデバイスドライバプロセスの生成要求を制御し，適当なタイミングで行う．ドライバ構成制御から要求を受けた適応制御機能はプロセス生成システムコールを発行し，プロセスを生成する．生成されたデバイスドライバプロセスは処理要求を受けるために情報を内コアに登録する．内コアは登録の可否をドライバ構成制御に問い合わせる．これにより，ドライバ構成制御を介して生成されたデバイスドライバプロセスのみの生成を許可し，(要求3)を満足することができる．また，内コアは登録をドライバ管理に通知する．ドライバ管理がデバイスドライバプロセスに対してprobe, attachの順序で初期化を行う．初期化の終了したデバイスドライバプロセスは動作可能になる．

(2) OS動作中に検出

OS動作中に検出するデバイスとしてはPnPデバイスを想定している．

PnP デバイスが接続されると，バスドライバが接続を検出し，デバイス情報をプログラム決定に通知する．その後は，OSの立ち上げ時と同様である．

5.2.2.　AP のデバイス利用要求を契機とした生成

AnTでは，必要に応じてドライバの起動，終了を行う．このため，OSに組み込まれていないドライバについては，入出力機器(デバイス)の利用要求を契機としてデバイスドライバを起動する必要がある．ここではAPからのデバイス利用要求を契機としたデバイスドライバプロセス生成の流れを説明する．処理の流れを図6に示す．

APは，ファイル管理に対して特殊ファイル名を用いてデバイスの利用を要求する．ファイル管理は，特殊ファイル情報表を利用して特殊ファイル名をデバイス番号に変換し，ド

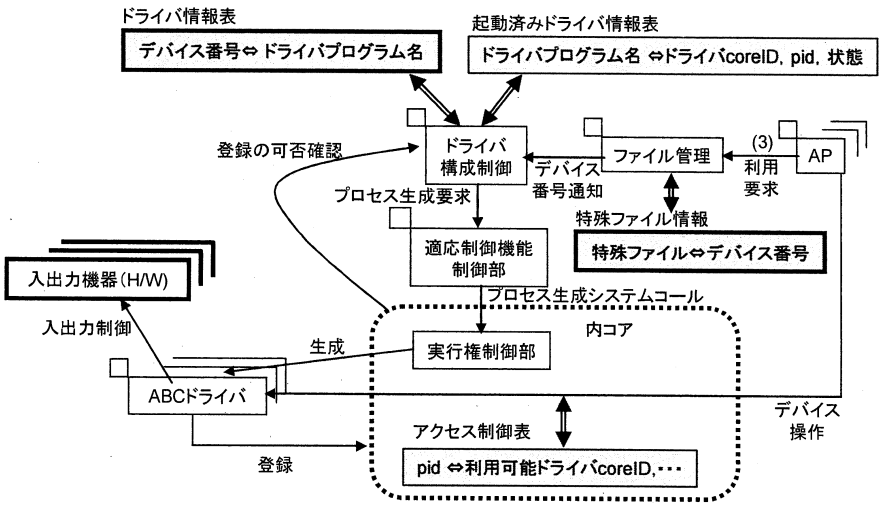

図 6 利用要求を契機としたドライバプロセスの生成の流れ

ライバ構成制御に対してデバイス番号を用いて利用要求を行う．ドライバ構成制御は，ドライバ情報表を利用してデバイス番号からドライバプログラム名を取得し，起動済みドライバ情報表を参照してデバイスドライバの状態を確認する．このとき，ドライバの状態としては，表3のような3つの状態がある．

表3　デバイスドライバの状態

状態名	状態
読み込み前	プロセスとして生成される前の状態
初期化前	プロセスとして生成されており，初期化を行う前の状態
動作可能	初期化が終了し，AP から利用可能な状態

状態が読み込み前である場合は，対象のデバイスドライバをプロセスとして生成する．デバイスドライバは，生成されると，内コアへ登録する．このとき，内コアはドライバ構成制御に対して，登録の可否確認を行い，ドライバ構成制御が起動したドライバの登録のみを許可する．登録の可否確認によりドライバ構成制御は，ドライバの初期化が終了したことを確認し，起動済みドライバ情報表の情報を更新する．ファイル管理からドライバ構成制御への利用要求の戻りの形で，ドライバ構成制御は，ドライバの識別子をファイル管理に通知する．ファイル管理は，要求元のプロセスとドライバの対応を内コアに通知する．内コアは，プロセスとデバイスドライバの対応関係をアクセス制御表へ追加する．また，APからファイル管理への利用要求の戻りの形で，ファイル管理は，ドライバの識別子を通知する．本処理により，内コアがAPのデバイス操作に対して利用要求を行なった外コアにのみアクセスを許すという制御が可能になる．

APは，取得したドライバの識別子を用いてデバイスを操作する．

5.3.　サービスプロセスの生成制御

サービスプロセスはコマンドインタプリタを用いたユーザの生成要求を契機として生成が行われる．また，***AnT*** ではサービスの要求に対して，そのサービスに深く関わるほかのサービスやドライバについても起動を行う連携機能を実現する．***AnT*** におけるサービスプロセス生成の様子を図 7 に示し，以降で説明する．

ユーザの入力によりコマンドインタプリタはプログラム名を取得し，プログラム決定部に通知する．プログラム決定部は通知されたプログラム名をデバイスドライバかサービスかを判別する．サービスの場合は適応性制御

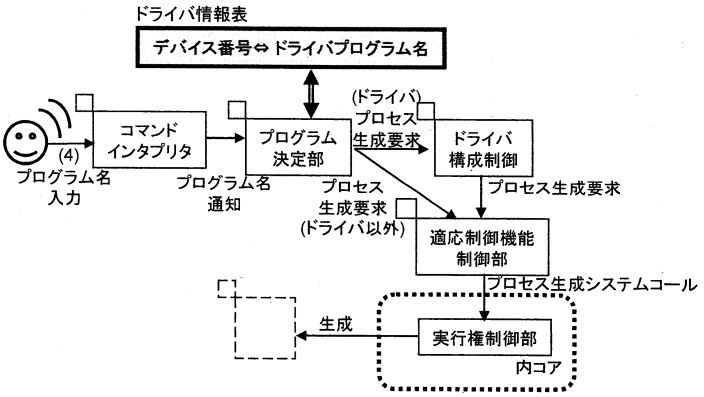

図 7　ユーザの生成要求を契機とした生成

にプログラム名を通知し，デバイスドライバの場合はドライバ構成制御にプログラム名を通知する．このとき，関連の深いプログラムのプログラム名も同時に通知する．適応制御機能は，通知されたプログラム名を用いてプロセス生成システムコールを発行し，実行権制御部がプロセスを生成する．

このように，デイバスドライバとサービスの生成要求を両方ともプログラム決定部に通知し同様に扱うことで，深く関連するサービスやドライバを連携して起動することができるようになる．

6. まとめ

AnT におけるプロセス生成制御機構について述べた．プロセス生成制御機構への要求として OS の動作に必須なプロセスの生成，システムに必要なプロセスの生成，デバイスドライバの起動制御，効率的なプロセスの走行があることを示した．プロセス生成制御機構の概要を述べ，効率的なプロセスの走行の実現を目指す適応制御機能について述べた．そして，プロセス生成の流れを OS 立ち上げ時，ハードウェアの検出時，AP のデバイス利用要求時，およびユーザからの生成要求時の 4 つに分類し，それぞれの生成の流れについて詳細に述べた．OS の立ち上げ時に，初期化プロセスにより OS の動作に必須なプロセスを生成し，ハードウェアの検出と AP のデバイスの利用要求という契機を用いてデバイスドライバの起動制御を行い，計算機の利用環境を把握し適切なデバイスドライバを起動することを示した．このとき，デバイスドライバの起動は必ずドライバ構成制御を介すようにし，デバイスドライバが内コアに登録をしたときにドライバ構成制御によって可否確認をすることで不用意なデバイスドライバの起動を抑制する．また，ユーザからの生成要求時にサービスの生成要求をデバイスドライバと同様にプログラム決定部に通知することで，関連の深いプログラムを起動する連携機能についても述べた．

今後は，***AnT*** に本機構を実装する予定である．

謝辞 本研究の一部は，科学研究費補助金 基盤研究(B)「適応性と頑健性を有する基盤ソフトウェアのカーネル開発」(課題番号：18300010)による．

参考文献

[1] 横山和俊，乃村能成，谷口秀夫，丸山勝巳，“応用プログラムの走行モード変更機構,” 情処研報，2004-OS-95，pp.25-32，2004.

[2] 仁科匡人，谷口秀夫，“プログラムの動作を利用履歴に適応させる適応制御機能,” 情処研報，2006-OS-103，pp.71-78, 2006.

[3] 谷口秀夫，乃村能成，田端利宏，安達俊光，野村裕佑，梅本昌典，仁科匡人，“適応性と堅牢性をあわせ持つ ***AnT*** オペレーティングシステム,” 情処研報，2006-OS-103，pp.95-102, 2006.

[4] 滝口真一，田端利宏，乃村能成，谷口秀夫，“ドライバプログラムの効率的な構成制御法,” 情報処理学会第 69 回全国大会講演論文集，第 1 分冊，pp.27-18, 2007.