情報処理 (計算機) システム Ⅱ ・第 3 回

2023 年 4 月 21 日

今回の内容

3.1 カーネル

オペレーティングシステムのカーネルは、1 つの (機械語) プログラムとして動作します。アプリケーションプログラムやサーバプログラムは、すべて、このカーネルから派生したプロセスとして起動されます。これらのプログラムを、カーネルプログラムと区別して、**ユーザプログラム**と呼びます。

ブートストラップ

PC の電源が投入されてからカーネルが起動されるまでの過程を**ブートストラップ**[1](bootstrap) と呼びます。PC の電源が投入されれると、マザーボード上の不揮発性[2]メモリ[3]に記憶されている (機械語) プログラム[4]が実行されます。このプログラムは、PC に接続されている入出力装置や補助記憶装置などを初期化し、マザーボードの設定[5]に従って、次に起動するプログラムを補助記憶装置から読み込みます。たとえば、SSD やハードディスクなどの場合、その記憶領域の先頭に置かれているプログラムを主記憶装置 (メモリ) 上に読み込み、実行します。このプログラムが、さらに (補助記憶装置に記憶された) より大きくて複雑な別のプログラムを主記憶装置上に読み込み、それを起動します。場合によってはこれをさらに繰り返すことで、補助記憶装置に記憶されているカーネルプログラムが主記憶装置に読み込まれ、カーネルが起動します。

メモ

[1]略して、単に、**ブート** (boot) と呼ぶ場合もあります。

[2]電源を切っても記憶が持続するメモリを不揮発性メモリと呼びます。一方、主記憶装置などに用いられるメモリは電源を切ると記憶が失われてしまいますが、このようなメモリを揮発性メモリと呼びます。

[3]この不揮発性メモリは**BIOS ROM** (Basic Input/Output System Read Only Memory) と呼ばれます。電源投入時にまず起動されるプログラムや、PC に接続された入出力装置の制御するための基本的なサブルーチン類のプログラムが、この BIOS ROM に記憶されています。

[4]このプログラムは Initial Program Loader と呼ばれます。

[5]主記憶装置とは別の揮発性メモリに PC のハードウェアの設定などが記憶されます。このメモリについては、PC の電源が切られている状態でも、ボタン電池などから電源を供給されるようになっていますので、その内容が失われることはありません。

補助記憶装置上に格納されているカーネルプログラムは、そのカーネルのファイル管理機能で作成できる 1 つのファイルとして存在するのが普通です。カーネルプログラムを主記憶装置にコピーして起動するプログラムは、当然、カーネルの助けなしで、カーネルプログラムのファイルを読み込まなければなりませんので、このファイルがどのように補助記憶装置上に格納されているのかに関する知識が必要です。また、この読み込みに必要な補助記憶装置の制御もカーネルの機能の一つですが、これもカーネルの助けなしに行う必要があります[6]。

メモ

3.2 カーネルの構成

カーネルプログラムは、前回説明した、プロセス管理、メモリ管理、ファイル管理、デバイス管理などの機能をそれぞれ担う多くの部分 (サブシステム) から構成されています。サブシステムは、カーネルプログラムの一部にはじめから組み込まれている場合もあれば、ハードディスクなどの補助記憶装置にファイルとして格納されており、必要に応じて主記憶装置上に読み込まれて動作中のカーネルプログラムと結合されることもあります。

カーネルが起動すると、CPU や主記憶装置を初期化するとともに、PC に接続された各入出力装置、補助記憶装置などを制御するためのサブルーチン群[7]が読み込まれます。カーネルの各サブシステムは、各種のデバイスの初期化などの必要な処理を行い、すべての準備が整った段階で、カーネルはいくつかのユーザプログラムをサーバプログラムとして起動します[8]。他のユーザプログラム (他のサーバプログラムやアプリケーションプログラム) はすべて、これらのサーバプログラムの働きで間接的に起動されます。

メモ

[6] カーネルプログラムを読み込む際の、補助記憶装置の制御は BIOS ROM に格納されたサブルーチンプログラムを呼び出すことで行われます。

[7] 特定のデバイス (入出力装置や補助記憶装置など) を制御するために使用されるサブルーチン群は、そのデバイスの**デバイスドライバ**と呼ばれます。多種多様なデバイスが存在し、それぞれにデバイスドライバが必要となるのが普通ですので、デバイスドライバは、起動時に必要に応じてカーネルに結合されるのが普通です。

[8] Linux カーネルの場合、カーネルは `init` という名前のサーバプログラムを起動します。すべてのサーバプログラムやアプリケーションプログラムは、この `init` から間接的に起動されます。

3.3 CPU の特権モードと非特権モード

計算機上の資源を管理するのがカーネルの役割ですが、この管理機構を外れて、ユーザプログラムが各種の資源に直接アクセスすることを許してしまうと、カーネルによる管理が破綻してしまいます。このため、PC やスマートフォンなどで利用されるような (ある程度高度な) オペレーティングシステムでは、カーネルが管理している資源にユーザプログラムが直接アクセスできないようにする仕組みが備わっています。

計算機上の資源へのアクセスは、CPU の特別な機械語命令を介して行われます。たとえば、メモリ資源を制御するには CPU のアドレス変換機構の設定を行う命令が必須ですし、デバイス管理では入出力装置へアクセスする特殊なデータ転送命令 (入出力命令) が必須となります[9]。PC やスマートフォンなどで使用されている CPU は、これらの特別な命令を**特権命令**と呼んで区別し、特権命令を実行できるモードと実行できないモードの 2 種類のモード[10]で動作することができるようになっています。前者を**特権モード**[11]、後者を**非特権モード**[12]と呼びます。

メモ

3.4 システムコール

特権モードで動作中のプログラムは、自由に非特権モードに切り替えることができますが、非特権モードから特権モードへの移行は自由にはできません。カーネルは特権モードで実行されますが、非特権モードに移行してから、ユーザプログラムの実行を開始します。ユーザプログラムは非特権モードでで動作しますので、各種の資源にアクセスするために必要な特権命令を実行することはできません。

ユーザプログラムが、各種の資源にアクセスしたい場合には、**ソフトウェア割り込み**[13]と呼ばれるものを発生される機械語命令によって、非特権モードから特権モードへ移行し、そこでカーネルに必要な処理を行ってもらいます。これを**システムコール (system call)**と呼びます。

ソフトウェア割り込みが発生した CPU は、カーネルが特権モードで動作しているときに設定した特定の手順 (サブルーチン) の実行を開始します。これにより、カーネルがあらかじめ登録しておいた特定の手順 (カーネルプログラムの一部) が特権モードで実行され、そこでユーザプログラ

[9]特定の物理アドレスへのアクセスを介して入出力を行う場合はアドレス変換機構を制御することで入出力装置へのアクセスを制御できます。

[10]CPU よってはさらに多くのモードを選択できる場合もあります。

[11]**カーネルモード** (kernel mode) と言ったり、**スーパバイザモード** (supervisor mode) と言ったりすることもあります。

[12]**ユーザモード** (user mode) と言うこともあります。

[13]割り込みについては次回に説明します。

ムが要求した操作がカーネルプログラムによって行われます。こうすることで、各種の資源をすべてカーネルプログラムの管理下に置くことが可能になります。

メモ

ユーザプログラムは、システムコールを行う前に、CPU の特定のレジスタ (群) に、カーネルに依頼したい処理の内容を指定する情報を格納しておき、カーネルがこの値を調べてユーザプログラムの要求の内容を識別します。これが機械語レベルでのシステムコールの呼び出しの仕組みです。

一方、多くの OS では、C 言語などの高級言語で書かれたプログラムから容易にシステムコールを行うことができるように、システムコールを行うサブルーチン群 (C 言語の場合は関数群) を納めたライブラリを提供しています。このようなライブラリをアプリケーションプログラムと結合することで、関数呼び出しの形でカーネルの提供する機能を使用することができます。このとき、一つのシステムコールが一つの関数に対応することもあれば、いくつかのシステムコールを組み合わせて、一つの関数が実現される場合もあります。

メモ

情報処理 (計算機) システム Ⅱ ・第 3 回・終わり