

■目次 (ほげが今回の範囲)

第1部 0と1だけでどうやって計算するの?	★9 コンピュータシステム
第2部 コンピュータの気持ち	★10 記憶装置
第3部 情報をどのように表現するか	★11 オペレーティングシステム
第4部 <u>コンピュータシステム</u>	⇒ ★12 ネットワークのしくみ

前回の授業では、現代のコンピュータシステムは、主にアクセス速度と価格の理由から、複数種類の記憶装置を使い分けていることを知った。今回の前半は、そのような構成の代表例として、CPUと主記憶の間に置かれるキャッシュメモリについて学ぶ。後半は、システムを構成する重要なソフトウェアであるオペレーティングシステムについて、何をしているのか、どんな機能をもっているのかを学ぶ。

★10 記憶装置 (承前)

★10.4 キャッシュメモリと記憶階層

これまで学んできたように、CPUはメモリ(主記憶装置)から機械語命令を一つずつ読みだして実行していく。命令によっては、さらに実行の際にメモリとの間でデータの転送を行うこともある。このようにCPUは頻繁にメモリにアクセスするので、アクセスに時間がかかる(☆1)ようだと待たされてしまい、命令を効率よく実行できない。現代のコンピュータシステムでは、CPUの動作速度(☆2)に比べてDRAMでできたメモリのアクセス速度(☆3)が遅いため、このことが大問題となっている。

Q1. クロック周波数が2GHzのCPUがある。このCPUは、メモリアクセスに全く時間がかからないという理想的な条件では、1クロック周期の間にちょうど1つの命令を実行できるという。この場合、このCPUが1つの命令を実行するのにかかる時間は何nsか。

上記の問題への対処のため、現代のコンピュータでは、主記憶装置よりもアクセス速度の速い記憶装置をCPUと主記憶との間に置いている(下記参照)。この記憶装置のことを**キャッシュメモリ**(☆4)(☆5)という。

☆1) レイテンシとスループット(第12回★8.1参照)のうち、この節での議論には主にレイテンシが関わっている。

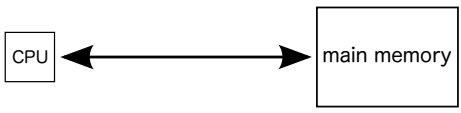
☆2) PCやスマートフォン向けCPUの場合、典型的なクロック周波数は1-4GHz程度。

☆3) 数十ns程度。第13回表1参照。

☆4) cache memory. cacheは、隠し場所、貯蔵庫、といった意味の語。現金を意味するcashとは別。

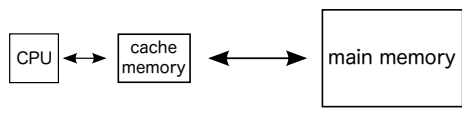
☆5) キャッシュメモリについては2年後期「計算機システムII」でより詳しく学べる。

● キャッシュメモリなし



CPUの動作速度に比べて主記憶へのアクセス速度は遅い。CPUはメモリアクセスのたびに待たされることになるので、効率よく命令を実行できない。

● キャッシュメモリあり



主記憶よりも高速にアクセスできるキャッシュメモリを用意し、何度も使う情報をそちらにコピーしておく。CPUの求める情報がキャッシュにあれば短時間でアクセスできる(ない場合は主記憶から読み出し、キャッシュにもコピーしておく)ので、メモリアクセス時間が平均として減少する。

CPU と主記憶の間に置くキャッシュメモリには、DRAM よりもアクセス速度の速い SRAM が用いられる。SRAM は DRAM よりも容量あたり単価が高いため、キャッシュメモリの容量は主記憶に比べてずっと小さい。現在の PC では、主記憶が数十 GiB の容量をもつものに対して、キャッシュメモリは数 KiB から数 MiB の容量のものが一般的である。CPU に内蔵されていることが多い(☆6)。

キャッシュと主記憶だけでなく、現代のコンピュータシステムでは、アクセス速度、容量、コスト(容量あたり単価)の兼ね合いで、種類の異なる記憶装置を組み合わせて使用している。CPU が高速にアクセスできるものほど容量が小さく、低速なものほど容量が大きくなっており、図1に示すように階層構造を成している(☆7)(☆8)。

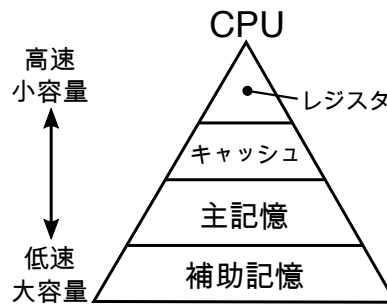


図1 記憶装置の階層

☆6) 以前はCPUとは別のICチップを使っていたが、近年ではCPUを構成する集積回路とあわせて一つのチップとなっている。第6回★5.5の図参照。

☆7) こにちのPC向けCPUでは、内蔵されたキャッシュメモリがさらに複数の階層に分かれているものが多い。それらは、CPUに近い側から、一次キャッシュ、二次キャッシュ、...、あるいはL1(Level 1)キャッシュ、L2キャッシュ、...などと呼ばれる。

☆8) HDDなどの補助記憶装置は非常に低速(第13回表1参照)なので、DRAMを「キャッシュメモリ」として内蔵しているものも多い。

★ 11 オペレーティングシステム

★ 11.1 オペレーティングシステムとは

第13回★9で学んだように、コンピュータシステムはハードウェアとソフトウェアで構成されており、そのソフトウェアは**アプリケーションプログラム**と**オペレーティングシステム (OS)**の2種類に分類される。OSは、図2に示すようにユーザおよびアプリケーションプログラムとハードウェアの間を仲介し、コンピュータシステムを便利で効率よく使えるように働く。

補助記憶装置のデータを読み書きするアプリケーションプログラム（以下 AP と略す）の例を考えてみよう。補助記憶装置には様々な種類・機種のものであるので、インタフェースの仕組みも様々である。APが自分でデータを読み書きするなら、個々のインタフェースに対応したプログラムをいくつも用意しなければならない。これは大変な手間である。入出力装置など、補助記憶装置以外のハードウェアでも同様である。OSは、このような手間を省く役割をしている。補助記憶装置等を直接操作する仕事はOSが請け負い、APはOSを介して情報のやりとりをするのである。こうすることで、個々のハードウェアの違いをOSが吸収して、APやユーザがその違いを気にせずに済むようにできる。

また、複数のAPが動作している場合、それらが同時に一つのハードウェアを使おうとしても混乱が生じないようにしなければならない(☆9)。ある入力装置からの情報を、本来受け取るはずのものとは別のプログラムが受け取ってしまう、ということも避けなければならない。このような問題が生じないようにハードウェアやAPの間を取り持つのもOSの役割である。

上記がOSの基本的な役割であるが、このような役割をはたすプログラムだけでなく、それを補助する様々なソフトウェア群もあわせて「オペレーティングシステム」と呼ぶこともある(図2の広義のOS)(☆10)。この場合のOSを構成する要素には、次のようなものがある。

カーネル 上述の最も基本的な役割をはたす、OSの中核部分。狭い意味では、OSとはこの部分を指す。

ライブラリ 汎用の処理を行うプログラムの「部品」を集めて、APなどが呼び出して利用できるようにまとめたもの(☆11)。

サーバプログラム 各種のサービスを提供する。システム上で常に動作し、APなどからの要求に回答して仕事をする(☆12)。

ユーティリティプログラム 主にユーザとやりとりして簡単な仕事をするプログラム(☆13)。

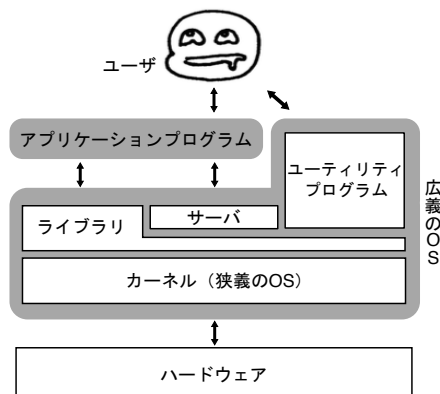


図2 オペレーティングシステムとアプリケーションプログラム/ユーザ/ハードウェアとの関係(☆14)。

☆9) 例えば、高橋がPCでゲームをしながら表計算ソフトでみなさんの試験の点数を集計してたとして、2つのプログラムがゲームのスコアと試験の点数を同時にHDDに書き込もうとしても、ごっちゃになってしまったりせえへんように。ひどいとえ...

☆10) Microsoft Windows, macOS, iOSは、広義のOSを指す名称と考えてよい。Linuxは、本来はカーネルを指す名称だが、広い意味で使われることも多い。Androidは、広義のOSに加えて標準的なアプリケーションプログラムまで1つにまとめたソフトウェア群を指す名称である。ちなみに、Androidのカーネルは、Linuxである。

☆11) C言語のprintfやscanfといった関数は、ライブラリを利用することでOSを介した入出力を行っている。

☆12) システムの状態を監視する、新たなハードウェアが接続されたことを検出して設定する、ネットワークを使うサービス(ウェブやメール)をする、等々。ちなみに、サービスを受ける側をクライアント(依頼人、顧客)という。

☆13) Linux等におけるlsやcdといったコマンドはその一例。他にも、ユーザからのキー入力を受け付けて、OSがこれらのコマンドやa.outを実行開始するように取り計らってくれる、**シェル**等がある。

☆14) この図は、中野先生の2014年度「情報処理の基礎」第14回資料の図をもとに作成した。

★ 11.2 オペレーティングシステムの機能

前節に述べたような OS (特にカーネル) の役割は、ひとことで「コンピュータシステム中の様々な**資源 (リソース)** を適切に管理すること」と言い表せる。ここでいう資源には、入出力装置や補助記憶装置などのハードウェアだけでなく、CPU の計算時間やメモリの使用量、補助記憶装置中のファイルといった抽象的なものも含まれる。プログラムはこれらの資源を使用して動作するが、資源には限りがあるので、無制限な使用を許すとシステムの効率が悪くなったり正常に動かなくなったりする。OS は、システムがそのような事態に陥らないように働いている。

上記のような OS の機能は、対象となる資源の種類によって次のように分類できる (ここではそれぞれの機能について簡単に説明するに留めます。詳しいことは、2 年後期「計算機システム II」で学べるでしょう)。

■**プロセスの管理** **プロセス** (☆ 15) とは、システム上で動作中のプログラムのことである (☆ 16)。現代のシステムでは、**マルチタスク** (☆ 22) という方式を採用している。マルチタスク方式のコンピュータシステムでは、複数のプロセスに CPU の計算時間を次々切り替えながら割り当てていくことで、プロセスたちを並行して動作させる。このようなシステムでは、1 つのプロセスがずっと CPU を専有して他に何もできなくなる (☆ 18)、重要な仕事をするプロセスがなかなか計算時間を割り当ててもらえずシステムの処理がもたつく、といったことのないように、OS がプロセスの実行を適切に管理する必要がある。

■**メモリの管理** プロセスは、自分のプログラムそのものや処理対象のデータ等を格納するためにメモリを必要とする。プロセスたちに割り当てるメモリ領域の場所やサイズを管理することも OS の大事な仕事である。これには、あるプロセスが間違っって別のプロセスのためのメモリ領域にアクセスしてしまう (☆ 19)、という事態を避けることも含まれる。

■**ファイルの管理** 現代のシステムでは、補助記憶上の情報を**ファイル**という抽象的な単位で扱うことが一般的である。OS は、AP やユーザが補助記憶にアクセスし易いように、このようなファイルやファイルを格納する**ディレクトリ** (☆ 20) などの構造とそれらを操作する手段を提供している。これを**ファイルシステム**という。ファイルシステムを構築・提供し管理することも OS の大事な機能である。

■**デバイスの管理** **デバイス**とは、コンピュータシステムのハードウェアのことである。前頁でも述べたように、入出力装置や補助記憶装置などの各種デバイスを管理するのも OS の大事な仕事である。これには、個々のデバイスの違いを OS 側で吸収して、AP やユーザからはその違いが見えないように／気にせずに済むようにすること (☆ 21)(☆ 22) も含まれる。

☆ 15) process. タスク (task) とも。

☆ 16) Microsoft Windows では、CTRL+SHIFT+ESC で起動する「タスクマネージャ」でプロセスを一覧できる。Linux や macOS などでは、ps コマンドや top コマンドで同様のことができる。

☆ 17) multi-task. マルチプログラミング (multi-programming) とも。PC 上で動画を視聴しながらワープロで文章を書いたりできるのも、マルチタスクのおかげ。

☆ 18) 自分の作った C 言語プログラムが無限ループしてずっと CPU を使い続け、キー入力もマウス操作も他のどんな処理も受け付けなくなる、なんてのは非常に困りますね。

☆ 19) 表計算ソフトの保持している試験の点数データがゲームソフトのスコアで書き換えられちゃったり、パスワードが他から見えちゃったり...

☆ 20) **フォルダ**ということもある。

☆ 21) これを、ハードウェア・デバイスを「抽象化」という。

☆ 22) 個別のデバイスを管理し、ユーザや AP に対して抽象化したインタフェースを提供するソフトウェアを**デバイスドライバ**という。OS に組み込まれて働くものが多い。