

OS 第 1 回課題レポート

1510151 柳 裕太

2017 年 6 月 6 日

1 課題 1: 和訳

1.1 OS のインターフェイス

オペレーティングシステム (以下、OS) の仕事は、コンピュータを複数のプログラムで共有したり、ハードウェアハードウェア単体のサポートより便利なサービスを提供することである。その OS は低級ハードウェアを管理・取り出しを行い、それにより、例えば、ワードプロセッサが自身のディスクハードウェアが使用中かどうか考慮する気にする必要がなくなる。多重のハードウェアもまた、多くのプログラムにコンピュータを同時に共有及び実行 (あるいは実行と見せかける) することを許している。最終的に、OS は制御された相互作用する方法を提供しており、それによってこれらはデータを共有できたり、あるいは共に仕事をすることができるのである。

単一の OS はユーザにインターフェースを介してプログラム群を提供する。よいインターフェースをデザインすることは難しいことがわかる。一方に、我々はインターフェースをより簡単に正しく実行するために、シンプルで精密なものにしたがる。もう一方に、我々はより洗練された特徴をアプリケーションに提供しよう誘惑されるかもしれない。この緊張を解くトリックは、インターフェースをもっと普遍的に提供できるようにするために、ほんの少しメカニズムに依拠するデザインにすることである。

この本は単一の OS を実態のある例として、OS のコンセプトを説明するために利用する。xv6 という OS は、ケン・トンプソンとデニス・リッチーによる UNIXOS の基本的なインターフェースを提供し、できるだけ UNIX 内部のデザインを模倣したものとなっている。UNIX はメカニズムもよく内包した限定的なインターフェースで、驚くべき汎用性を提供する。このインターフェースはにおいて成功してきており、現代の OS —BSD, Linux, MacOS, Solaris, そして更に、限定的では在るが、Microsoft Windows も—は UNIX のようなインターフェースを所持している。xv6 を理解することは、これらのシステムやその他を理解するためのよいスタート地点となるのである。

図 0-1 に記載されたとおり、xv6 は伝統的な kernel の形式をとっており、プログラムを走らせるためのサービスの特別なプログラムがある。それぞれの走っているプログラム (プロセスと呼ぶ) は、指示/データ/スタックを内蔵したメモリを保持している。この指示はプログラムの計算が実装されている。データは計算における変数である。スタックは、プログラム処理のコールが構成されてある。

プロセスがカーネルサービスを呼び出す必要がある時、まず OS のインターフェース内にて手続きを呼び出す。この手続きのことを”システムコール”と呼ぶ。このシステムコールがカーネル内部

に入り、カーネルがサービスと結果返しを行う。それゆえプロセスはユーザスペースとカーネルスペースとの処理のやりとりを代行しているのである。