

日曜研究室

技術的な観点から日常を綴ります

[xv6 #9] Chapter 1 – The first process – Address space overview

テキストの18～19ページ

本文

それぞれのプロセスにそのプロセス専用のメモリの見え方（アドレス空間と呼ばれる）を提供するために、xv6はページングハードウェアを使用する。

プロセスごとに個別に、プロセスのアドレス空間を定義するページテーブルを、xv6は維持する。

アドレス空間は、プロセスの仮想アドレス0から始まるユーザメモリを含む。

アドレス空間の中で、通常、命令は先頭にあり、次にグローバル変数とヒープ領域（mallocとかで使う）がある。

プロセスは、必要に応じてそのアドレス空間を広げる事ができる。

どのプロセスのアドレス空間も、ユーザのプログラムのメモリと同じように、カーネルの命令とデータを対応付ける。

システムコールによってプロセスが呼び出されたとき、そのシステムコールは、プロセスのアドレス空間のカーネルマッピングの中で実行される。（ちょっと意味が分からない。）

この決まりごとは、カーネルのシステムコールのコードが直接ユーザのメモリを参照出来るようにするために、存在する。

ユーザのメモリを展開する余地を残すために、xv6のアドレス空間は、0x80100000から始まる高いアドレスにカーネルを対応付ける。

感想

正直かなりあやふやです。

ただここはまだ概要で、次から詳細に入っていくので、そこで多分理解出来るんじゃないかなと思うってます。

現時点では、プロセスごとの仮想的なメモリ空間はかなり広く、その空間のはじめの方からプログラ

ムの命令やデータが置かれ、後ろの方にカーネルの命令やデータが置かれるんだと思ってます。

(32ビットなので4GB。ただし0x80100000以降はカーネルが置かれるので実質約2GBぐらいしか使えない?)

なんでそうなるのか(特にカーネル部分まで含む理由)はまだ分かりません。

ただプロセスごとにカーネルのデータを持つのは非効率なので、物理メモリ上でカーネルが載ってる同じ部分を指すようになってるんだと想像します。

さてこの想像(予想)は当たるか外れるか...

カテゴリー: 技術 | タグ: xv6 | 投稿日: 2012/2/14 火曜日 [<http://peta.okechan.net/blog/archives/1263>] |
