

日曜研究室

技術的な観点から日常を綴ります

[xv6 #13] Chapter 1 – The first process – Physical memory allocation

テキストの22ページ

本文

カーネルは、ページテーブル・ユーザメモリの処理・カーネルのスタックやパイプやバッファのために、実行時に物理メモリを割り当てたり解放したりする必要がある。

xv6は、その実行時割り当てにカーネルの終わりからPHYSTOPの間の物理メモリを利用する。

割り当てと解放は、4096バイトごとに行われる。

ページそれ自身をリンクリストに追加することで、どのページが空いてるかを記録する。

（ここかなり訳があやふや。原文: It keeps track of which pages are free by threading a linked list through the pages themselves.）

割り当ては、リンクリストからページの消去を行う。

解放は、そのリストに解放されたページの追加を行う。

起動中の問題がある。

物理メモリの全ては、アロケータのための空きリスト（前段のリンクリストの事）を初期化するためマップされていなければならない。

しかし、そのマッピングを使ってページテーブルを作成すると、ページテーブル用のページの割り当てを巻き込むことになる。

xv6はこの問題を、起動の間、カーネルのデータセグメントのちょうど終わりから割り当てるような、別のページアロケータを使うことによって解決している。

このアロケータは、解放をサポートせず、entrypgdirの4MBマッピングに制限される。

しかし、これは最初のカーネルのページテーブルを割り当てるのに十分である。

感想

実行時はどのページが空いてるかをリンクリストで管理するようです。

アロケータのために空きリストを用意しなきゃいけないのに、その空きリストのための領域をアロケータで割り当てようとする、要はニワトリが先かタマゴが先かみたいな話になってしまうってことですかね。

だから先に、簡易的なアロケータを使って解決しましたよってことでしょうか。

(entrypgdirは前々々回が出てきましたね。)

正直ピンとこないですが、次のCode: Physical memory allocatorを読めば理解が深まることを期待。

カテゴリー: 技術 | タグ: xv6 | 投稿日: 2012/2/18 土曜日 [<http://peta.okechan.net/blog/archives/1279>] |
