

コンピュータ アーキテクチャ I 第12回

塩谷 亮太 (shioya@ci.i.u-tokyo.ac.jp)

東京大学大学院情報理工学系研究科 創造情報学専攻

課題の解説

課題 1 1

- 以下のような状況の仮想メモリについて考える：
 - 仮想アドレス空間と物理アドレス空間は共に 32bit である
 - 単段ページ・テーブルを使用
 - ページ・サイズは 64KB である
 - ベース・レジスタには物理アドレス 0x20000000 が設定されている
 - 仮想アドレス 0x10000000 と 0xfea50000 から始まるページには, それぞれ物理アドレス 0x30000000 と 0x30010000 から始まるページが割り当てられているものとする
 - TLB は存在しない
- (すいません, 0xfea50000 のところが間違っていました)

課題 1 1

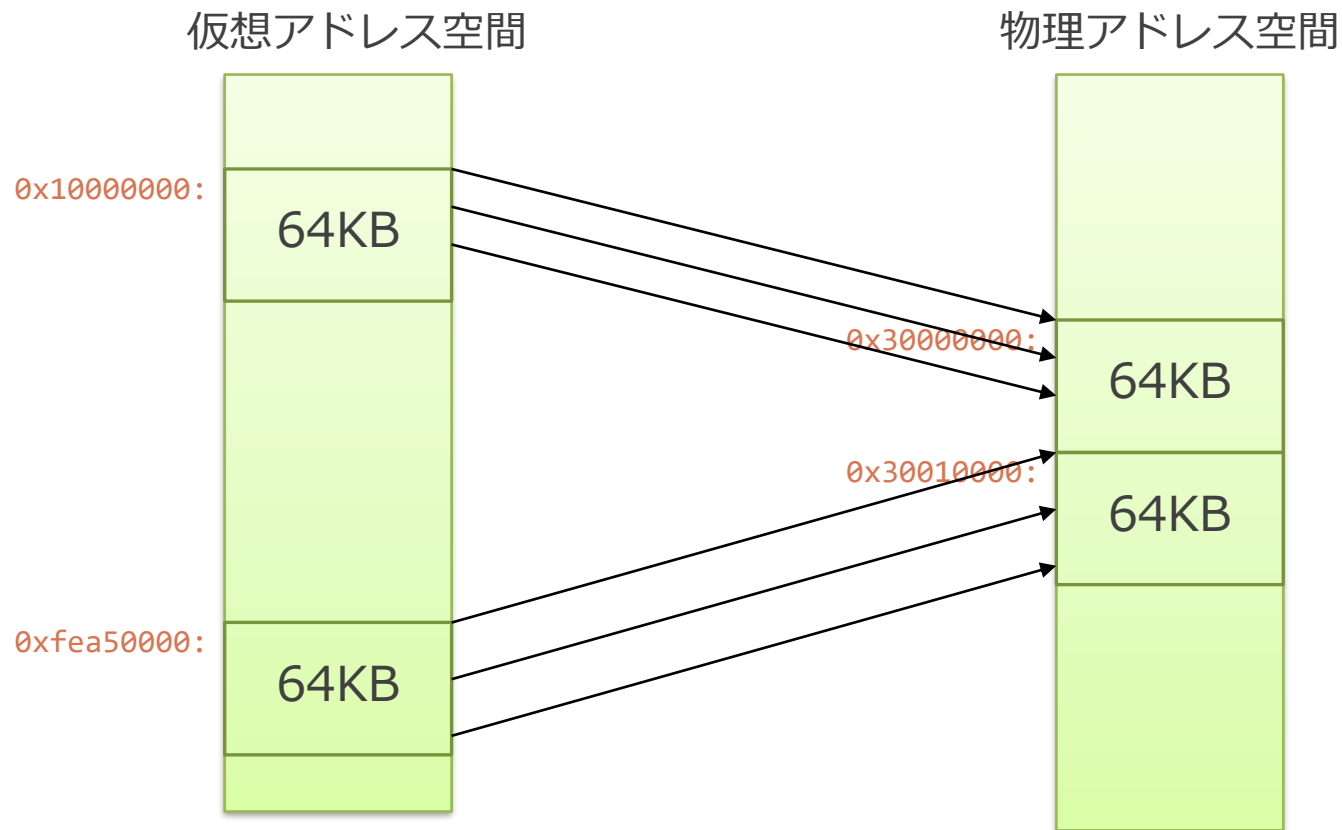
■ 前提からわかること

- ページサイズは $64\text{KB} = 65536\text{B} = 0x10000\text{B} \rightarrow$ 16進数で上4桁がページ番号
- アドレス空間は 32bit = ページ先頭をさすポインタは 4B

■ (1) 仮想アドレス $0x10002000$ に格納されている値を読み出す際にアクセスされる物理アドレスをすべてあげよ

- $0x2000\ 0000 + 4\text{B} * 0x1000 = 0x2000\ 4000$ に最初にアクセス
 - ここから $0x3000\ 0000$ が読める
- $0x3000\ 0000 + 0x2000 = 0x3000\ 2000$ に次にアクセス

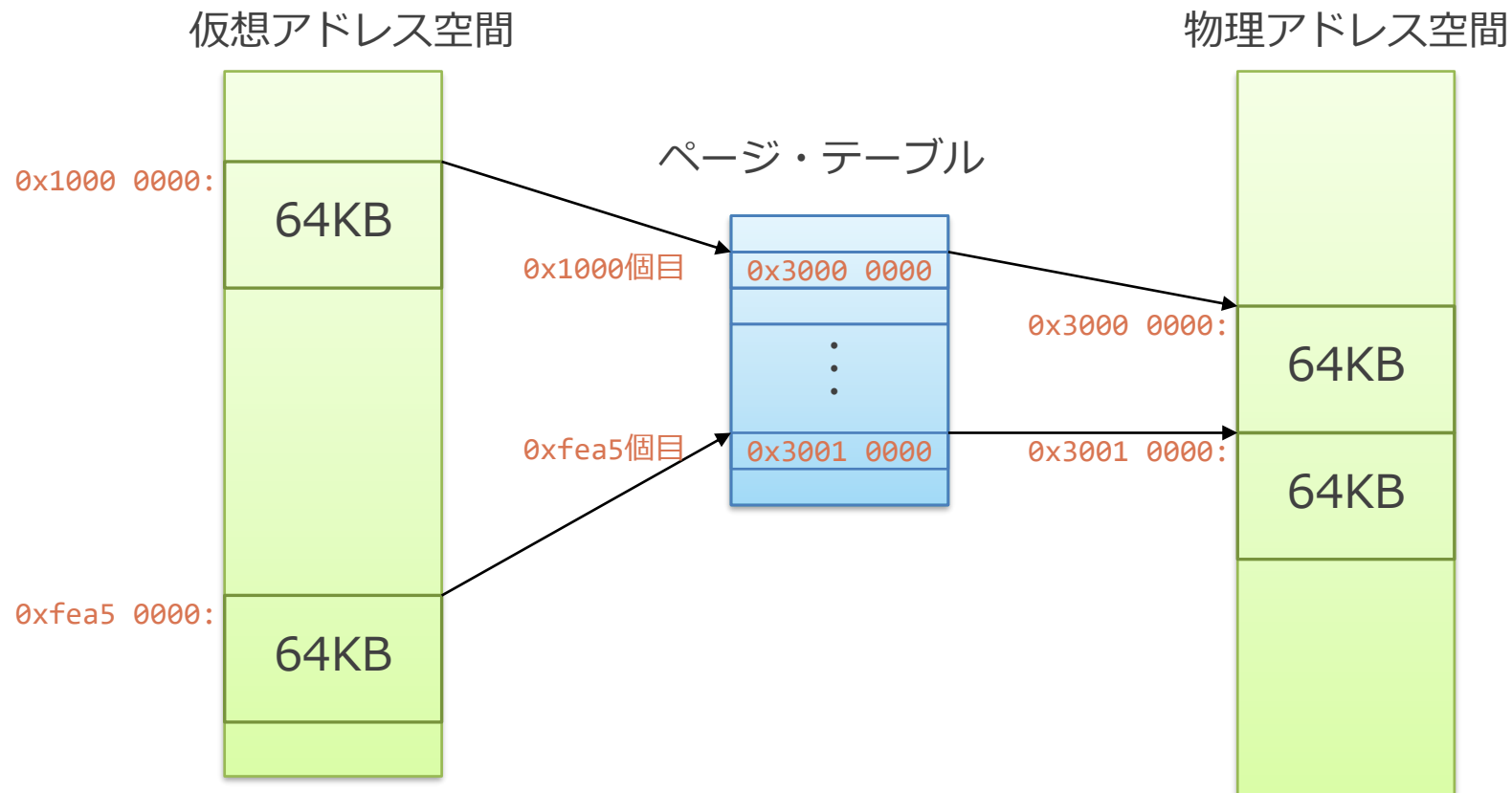
仮想メモリと物理メモリの対応



- 仮想アドレス上の 64KB が物理アドレスの 64KB のブロックにマップされている

- $64\text{KB} = 65536 = 0x10000$
- $0x10000000 \rightarrow 0x30000000$, $0x10000001 \rightarrow 0x30000001$, $0x10000002 \rightarrow 0x30000002$...
 $0x1000ffff \rightarrow 0x3000ffff$ にマップされている
- $0xfea50000 \rightarrow 0xfea50000$, $0xfea50001 \rightarrow 0xfea50001$, $0xfea50002 \rightarrow 0xfea50002$...
 $0xfea5ffff \rightarrow 0xfea5ffff$ にマップされている

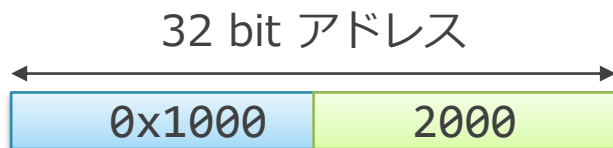
仮想メモリと物理メモリの対応



■ この対応付けを，ページテーブルによって表現する

- 64KB= $0x10000$ 個のブロック単位で仮想アドレスから物理アドレスに対応付けられている
- ページテーブルはポインタの配列であり，N番目の要素=仮想アドレスのN個目のブロックのためのポインタを格納
- 仮想アドレスから，その仮想アドレスが何個目のブロックなのかは $0x10000$ で割れば良い
 - $0x10000$ で割る = 16進数で下4桁（下16bit）を取りのぞく → 上16ビットが残る

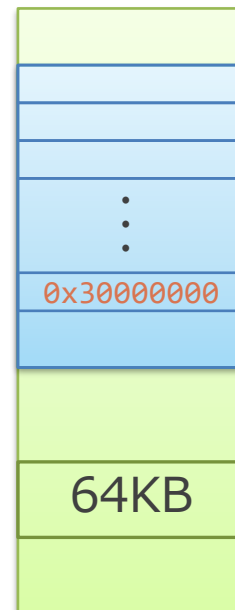
単段ページ・テーブルの動作



物理メモリ

0x20000000:
0x20000004:
0x20000008:

ページ・テーブル



■ ページ・テーブルは実際には物理空間上におかれる

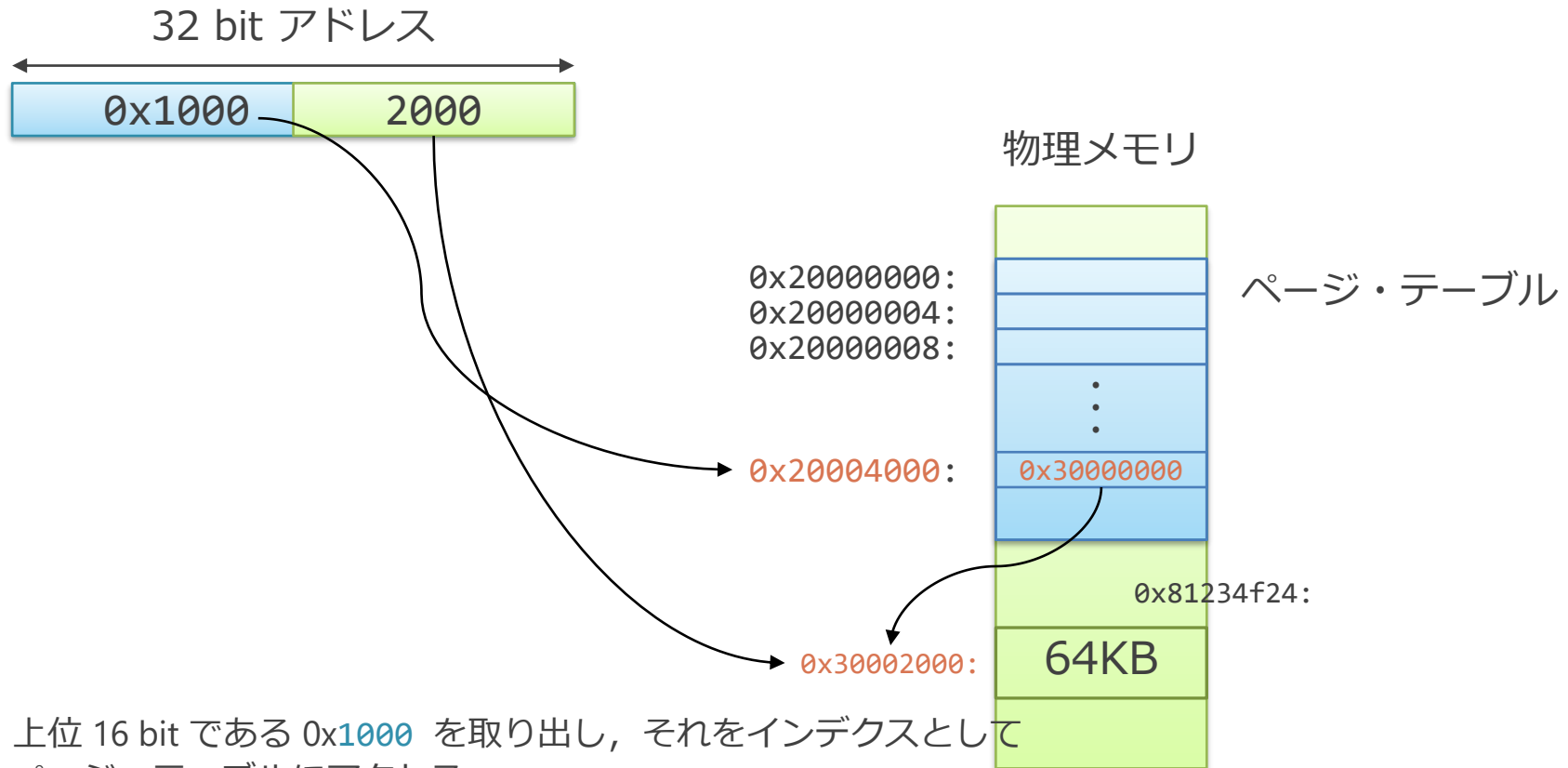
- その先頭アドレスがベース・レジスタで指定する
- 今回は 0x20000000 からがページ・テーブル

0x20004000:

■ 仮想アドレスからの変換の手順

1. 仮想アドレスをページ・サイズで割り、それがアドレス空間中にある N 番目のページなのかを計算する
2. ページ・テーブルの N 番目の要素を読む
 - $0x20000000 + N \times 4$ の物理アドレスを読む
 - *4 は格納されているポインタのサイズが 4B だから
3. 仮想アドレスをページ・サイズで割った余りを 2. で得られたアドレスに足す
 1. 余り=64KB ページの中の何番目のアドレスなのかを意味する
 2. 16進数で下4桁を取り出すことは、ページ・サイズ0x10000で割った余りを求めるのと等価 7

単段ページ・テーブルの動作



- 上位 16 bit である 0x1000 を取り出し、それをインデクスとしてページ・テーブルにアクセス
 - 物理メモリへのポインタはここでは 4B 単位であり、ベースレジスタは 0x2000 0000 をさす
 - $0x2000\ 0000 + 4B * 0x1000 = 0x2000\ 4000$ にアクセス
- マップされているページの先頭の物理アドレス 0x30000000 を得る
 - OS がこの論理アドレスに割り当てたページのアドレス
- 下位 16bit である 0x2000 と結合して 0x30002000 にアクセス

課題 1 1

- (2) 仮想アドレス `0xfea51fff` に格納されている値を読み出す際にアクセスされる物理アドレスをすべてあげよ
 - $0x2000\ 0000 + 4B * 0xfea5 = 0x2003\ FA94$ に最初にアクセス
 - ここから `0x3001 0000` が読める
 - $0x3001\ 0000 + 0x1fff = 0x30011fff$ に次にアクセス

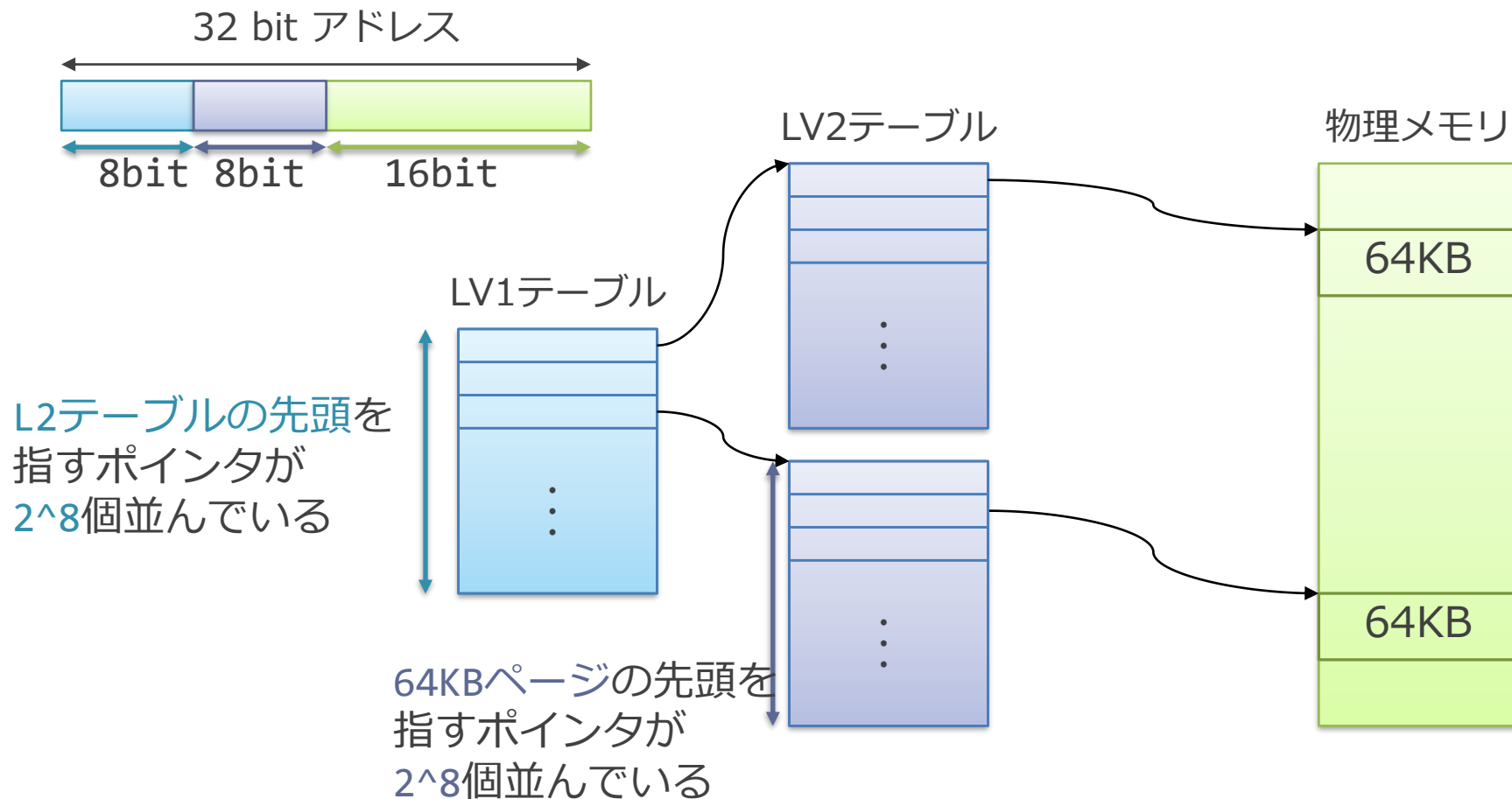
課題 1 1

- (3) 仮想アドレス $0x10000000$ と $0xfea50000$ から始まる2つのページのみが確保されている場合を想定する。この時に使用される物理メモリの容量の合計（ページ・テーブルとページそのもの）を求めよ
 - 単段テーブル全体 : $0x10000 * 4\text{Byte} = 0x40000 \text{ Byte}$
 - ページは $0x10000$ 個存在する
 - 確保されているページは2つ : $0x10000 \text{ Byte} * 2 = 0x20000 \text{ Byte}$
 - $64 \text{ KB} = 0x10000 \text{ Byte}$ が2つ
 - 合計すると $0x60000 \text{ Byte}$

課題 1 1

- (4) (3)と同様の条件で、2 段ページ・テーブルを使用した場合に使用される物理メモリの容量の合計を求めよ。この時 LV1 と LV2 に使用されるアドレスのビット幅は等しいものとする。
 - $256 * 4B * 3\text{テーブル} = 0xC00 \text{ Byte}$
 - 確保されているページは 2 つ : $0x10000 \text{ Byte} * 2 = 0x20000 \text{ Byte}$
 - 合計で $0x20C00 \text{ B}$

2 段ページ・テーブルの例



- 各レベルには $2^8=256$ 個のポインタを含むテーブルがある

期末試験について

- 8/7 いつもの講義の時間にこの部屋で
- A4 裏表 1 枚 手書きのみの持ち込み可
- 基本的に課題で出した部分を中心に出题する予定

- 質問がある場合は、遠慮無くメールで聞いて下さい
 - できるだけこの時間中に聞いて貰えるとよいです
 - 今週は出張に出ているため、やや反応が悪いかもしれないです
- 練習問題はまだ未完成ですが、講義資料の隣にアップロードしています

質問とか感想

- 35ページ計算のページアドレスの8Bはどこからわかるのかがわからなかったなので教えていただきたいです。

変換の実装

■ 単純な実装

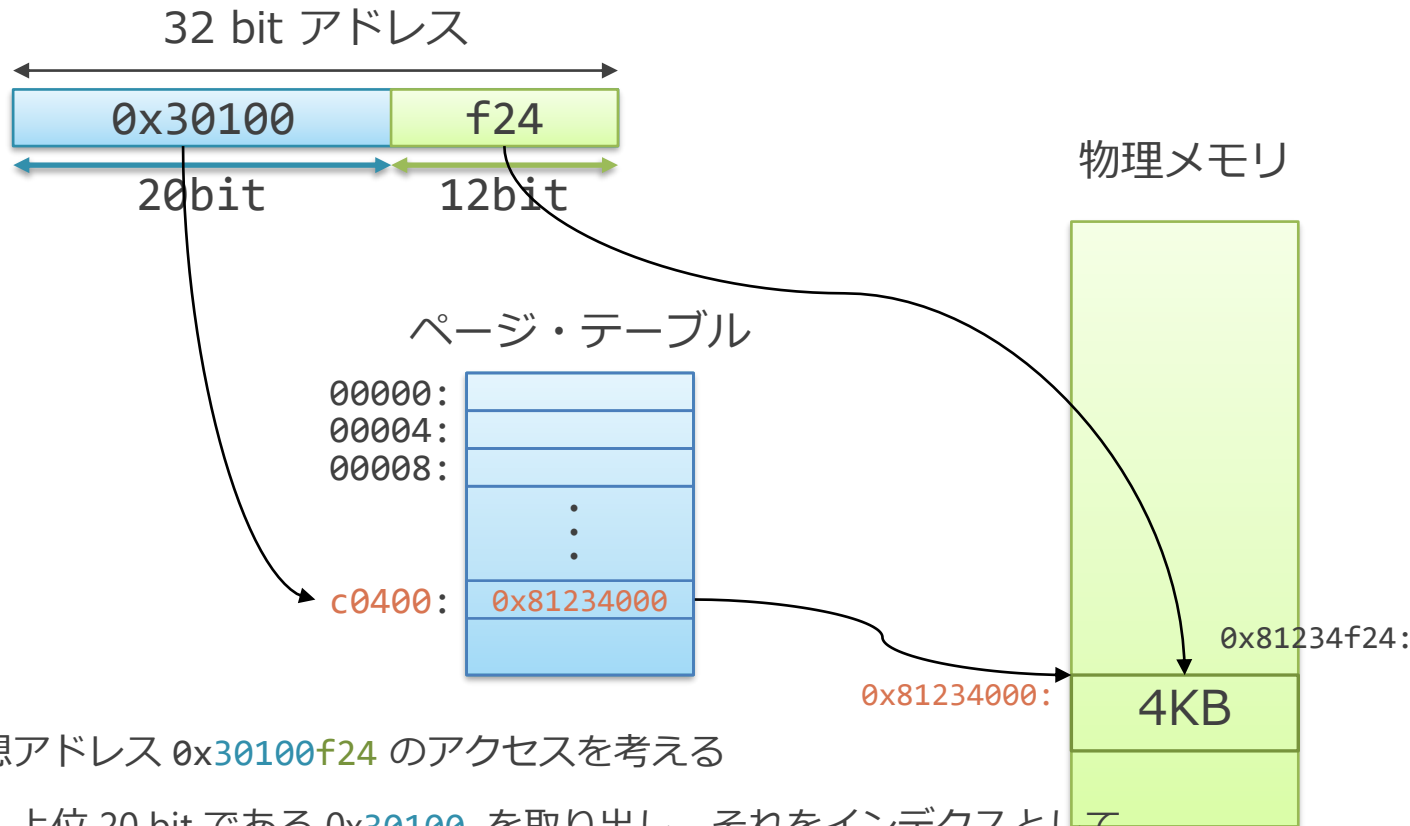
- 仮想アドレス → 物理アドレス の変換表を用意すれば良い

■ コスト：アドレスが 32 bit だとして, 1 byte 単位で表を作ると...

- データ 1 byte ごとに, その表には 32 bit = 4 byte のアドレスを記録することになる
- 実データの 4 倍の容量が変換表に必要になってしまう

- p38に、物理メモリへのポインタは4KBごとと書いてありますが、4KBごとではないのですか。

単段ページ・テーブルの動作



■ 仮想アドレス 0x30100f24 のアクセスを考える

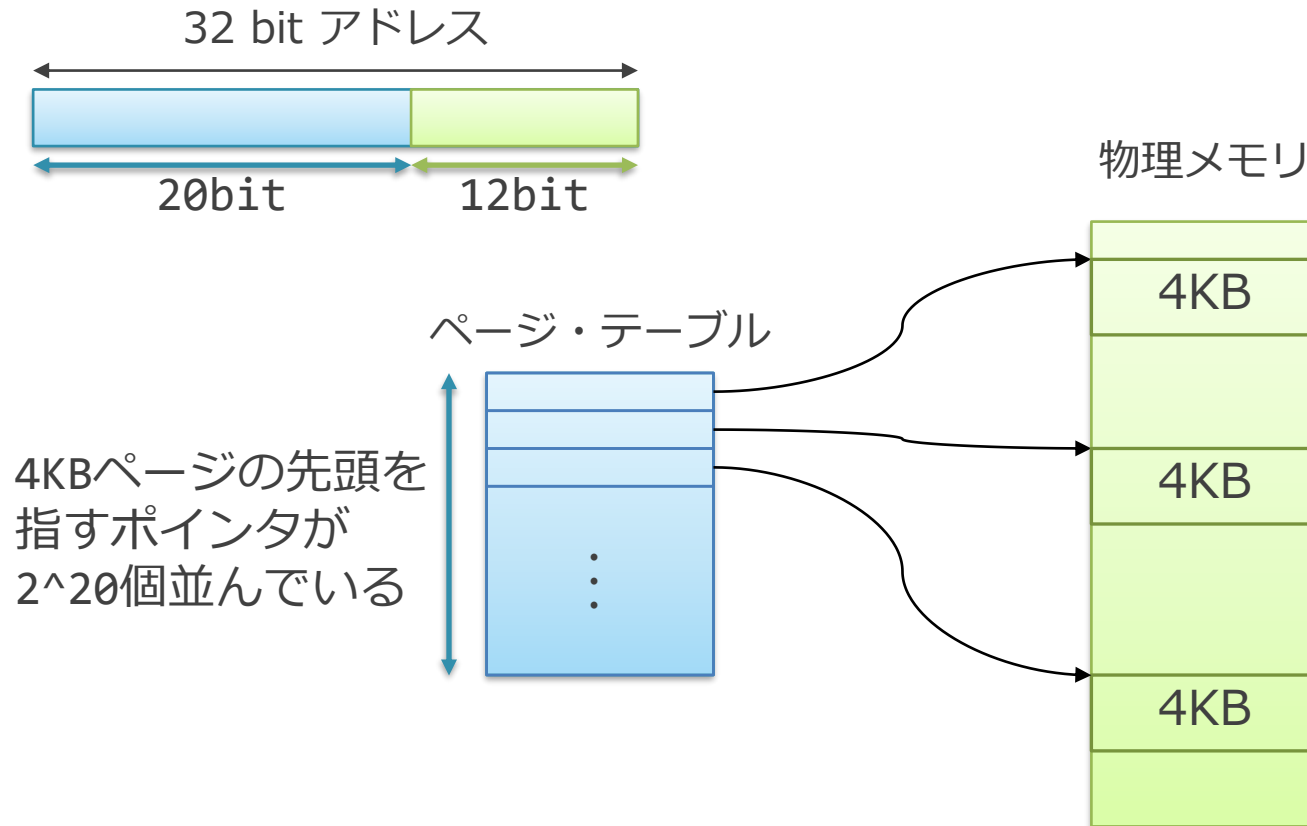
- 上位 20 bit である 0x30100 を取り出し、それをインデクスとしてページ・テーブルにアクセス
 - 物理メモリへのポインタはここでは 4B 単位なので、
 - $0x30100 \times 4B = 0xc0400$ にアクセス
- マップされているページの先頭の物理アドレス 0x81234000 を得る
 - 0x81234000 は OS がこの論理アドレスに割り当てたページのアドレス
- 下位 12bit である f24 と結合して 0x81234f24 にアクセス

質問とか感想

- 何度も目にしてきたSegmentation Faultが、カーネル・モードでは読めるがユーザー・モードでは読めないという設定から発生していると知ることができました。特権モードによりメモリを保護することができることが分かりました。
- カーネルモードで読める以外に、単に何も割り当てられていなくて読み出し禁止になっていることもあります

- 講義資料の37ページではポインタが 2^{20} 個並んでいると書いてあり、41ページの64bitのアドレス空間でページサイズを4KBとした場合では、アドレスの個数が 2^{64} と書いてあるのですが、何で決まっているのでしょうか。

4KB ページを使った単段ページ・テーブルの例



- まず単段のページ・テーブルを考える
 - アドレス・サイズが 32 bit, ページ・サイズ $2^{12}=4$ KB を仮定

多段ページ・テーブル

- ページ単位で管理したとしても, なおページ・テーブルは大きい
 - 64 bit のアドレス空間で, ページ・サイズを 4KB とした場合,
 - $(\text{アドレスの個数}) / (\text{ページ・サイズ}) * (\text{アドレスのサイズ}) = (2^{64}) / 4\text{KB} * 64\text{bit} = 16\text{EB} / 4\text{KB} * 8\text{B} = 32\text{PB}$
 - たとえ 1B しかメモリを使わないプログラムでも 32PB が必要に
 - 32 bit のアドレス空間なら大分でしたが, それでも 4MB が必要
- 多段ページ・テーブルと呼ぶ構造で効率良く保持する
 - プログラムで使うメモリ容量に比例した程度の容量でページ・テーブルを作る方法

質問とか感想

- 仮想アドレス空間から物理アドレスに変換する際、容量オーバーで割り当てることができない場合もsegmentation faultになるのでしょうか？
- たぶん違う名前のエラー（メモリ割り当てができなかったことを示す）がでると思います

質問とか感想

- ・ アドレスの個数をどうやって計算するかわかりませんでした。
- ・ スライド32ページから、アクセスするのは、例えば(1)なら「0x10004000」と「0x30006000」だと考えましたが、課題の問題文にある物理アドレスとは「0x30004000」と「0x30006000」だとも思い、「アクセスされる物理アドレス」とは何を指すのかわかりませんでした。
- ・ スライド35ページの「アドレスのサイズ」とは仮想アドレス空間のサイズか物理アドレス空間のサイズか、合わせたサイズなのかわかりませんでした。

- 35ページの容量の合計の計算の、「64bitのアドレス空間で」というのは仮想アドレス空間と物理アドレス空間のどちらかを表しているのか、それとも両方を合わせたものなのかどちらですか。

多段ページ・テーブル (P.35 じゃなくて P.41?)

- ページ単位で管理したとしても, なおページ・テーブルは大きい
 - 64 bit のアドレス空間で, ページ・サイズを 4KB とした場合,
 - $(\text{アドレスの個数}) / (\text{ページ・サイズ}) * (\text{アドレスのサイズ}) = (2^{64}) / 4\text{KB} * 64\text{bit} = 16\text{EB} / 4\text{KB} * 8\text{B} = 32\text{PB}$
 - たとえ 1B しかメモリを使わないプログラムでも 32PB が必要に
 - 32 bit のアドレス空間なら大分まだが, それでも 4MB が必要
 - $\text{アドレスのサイズ} = \text{アドレスを表すポインタのサイズ}$
- 多段ページ・テーブルと呼ぶ構造で効率良く保持する
 - プログラムで使うメモリ容量に比例した程度の容量でページ・テーブルを作る方法

質問とか感想

- スライドなどで解説が難しい場合、メールで教えていただきたいと思います。(質問の重複を避けるためメールは送っていません。)
- 重複しても構わないので、メールで送って下さい

- 講義資料27ページの、「メモリがスワップのキャッシュになっている」というのは、スワップ領域を普段のメモリに、メモリをキャッシュに例えているという解釈で正しいでしょうか？
 - そうです

- また、あるプログラムが他のプログラムのメモリの空き領域を使わせてもらうことで発生してしまうであろうバグが(p27)、仮想メモリを使うことで防ぐことができるということに疑問を持ちました。仮想メモリは全てのプログラムが仮想メモリ上での空き領域を使うのではないかと解釈したためです。

- c言語でシステムコール関数を利用する際いつもメモリを直接触っている感覚だったのですが、実際は裏でos が操作を行なっていることに驚きました。

- プログラミングの授業でシステムコールについて触れたので、とても面白かったです。mallocをmmapを使って実装しようとしたことがあるのですが、結構危ない作業だなあと再度思いました。

- テストが不安です、、、

- (3)のアドレスの個数はどのように計算したのですか。35ページの計算例では、2のアドレス空間のサイズ乗だったけれど、31ページの図ではページテーブルの大きさ乗だったので意味がわかりませんでした。

- 2段ページテーブルになったときの公式の変化がわかりません。アドレスの個数とアドレスのサイズが数値的にどれを使って何を表すのかがわかりません。ページサイズは変わらないと思っています。

- 実際のメモリには適当にマップされているから、物理アドレスから仮想アドレスを探すときに結局細切れになっていて探すのに時間がかかりそうだなと思いました。

- ページテーブルもパイプラインプロセッサも考え方が似ていて、コンピュータは同じような考え方で効率性を図ったり、無駄を無くしたりすることが多いのかなと思いました。

- バイトとbitが入り混じった時の計算方法がわかりません。

質問とか感想

- お忙しいとは思いますが、せめて演習課題の完全な解説を口頭ではなく文字や図として起こしていただければ試験勉強の目処が立ちます。よろしくお願いします。厳しいようでしたら試験ではなくレポートでの成績付与にしている教科もあるようですので無理のない範囲で検討していただければと思います。
- こちらは解答に必要な情報は全てのせているつもり（講義資料本体とあわせてみると）だけど、ここがわからないというのがあれば加筆するので言って欲しいです

- mmapについての説明が、今回のスライド以外を探してもなかったなので詳しく説明が聞きたいです。

- 物理メモリのポインタから物理アドレスを得るとき、その物理アドレスをどのように求めれば良いのか分かりませんでした。

- 特権モードのユーザーとカーネルのモードは、Windowsである「管理者権限で実行」と関係ありますか？
 - 関係しています。来学期に解説されるはず

- ページテーブルは単段のときに並んでいるポインタはいつも 2^{20} 個ですか？ またなぜ 2^{20} 個なのでしょう。
- アドレス空間全体にページが何個入るかで決まります
- この場合は、空間全体 $2^{32} / 4\text{KB} = 2^{20}$

質問とか感想

- 多段ページにすることは、図書館の本の分類と似ているようにも感じました。

（全部の本を一つの索引にまとめるのではなく、まずは大分類で振り分けてから小分類に分け、最後に小分類ごとの各本の索引を作る）

しかし、多段ページにしても本の総数が変わっているわけではないので、索引の数が減るということをいまいち実感できませんでした。（むしろ分類の数だけ余計に索引が必要）

ページ総数というより、アクセスする過程を減らすことに意義があるのでしょうか？

- 「か行」と「な行」から始まる本はいま存在しないので、「か行」と「な行」用の詳細な索引は省略できるイメージ

- ベースレジスタを用いる時とページテーブルを用いる場合の違いはなんですか？講義資料の32ページと33ページで同じ仮想アドレスへのアクセスを考えているのにどちらを用いるかによって得られる物理アドレスが変わってきてしまうように思いました。課題の時にもベースレジスタを使った時と使わないでページテーブルのみで考えた時で答えが変わる気がします。（課題の条件分5つ目はページテーブルのことだと思っています。勘違いしていたらすみません。）考え方の根本から違っていたら教えていただきたいです。

質問とか感想

- 猫の顔文字可愛すぎたしとてもわかりやすかったです!!!

一学期間おつかれさまでした
テストがんばってください

^^
(,,Д)
(っ っ

^_^
('▽`)
(っ っ

^^.
(*—`)
(っ っ

^_^
('ω`)
(っ っ