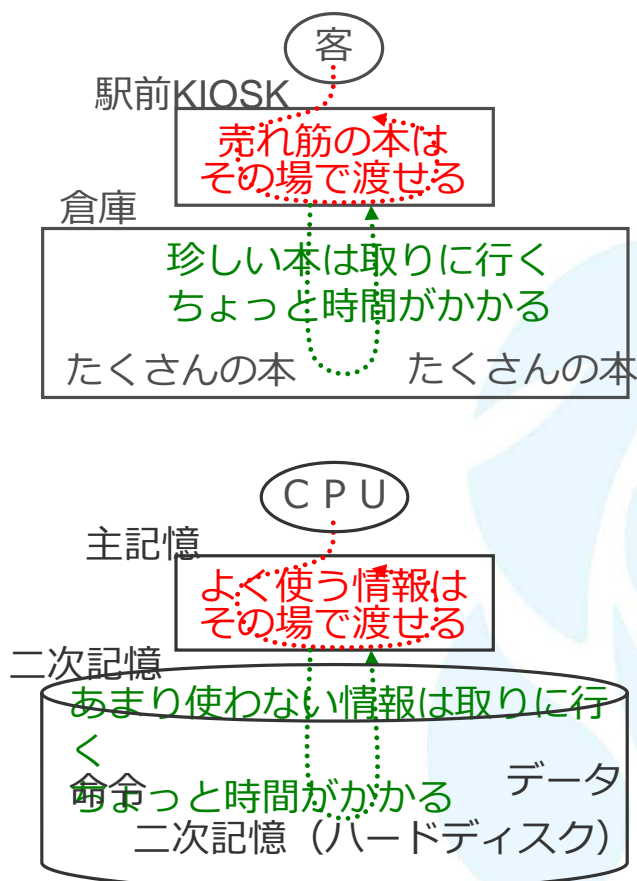


ページ置き換えの 動作とアルゴリズム まずは例 1 ~ F I F O

(復習) KIOSK本屋さんのたとえ話



- 売れ筋の本がKIOSKにあるためには、
倉庫からN冊まとめて
持ってきて置いておく
←次の客にも渡せる
- よく使う情報が主記憶にあるためには、
二次記憶からブロック
(ページ単位)で持って
きて置いておく
←次の要求にも渡せる

「置いておく」 ⇒ 段々あふれる

- 次の要求にも、すぐに渡せるように、もって来たページを主記憶に置いておく

「置いておく」 ⇒ 段々あふれる

- 次の要求にも、すぐに渡せるように、もって来たページを主記憶に置いておく
- ⇒ 主記憶にどんどん持ってくる
- ⇒ 段々あふれてくる
次に別のページを持ってこようとしても入れられる「空きページ」がない

「要らないページ」って何か？

- 「要らないページを追出す」というのだが、
「要らないページ」は何か？ どれを選ぶか？

(逆に) なぜ取っておきたいのか

～ ～ ～ ～ ～ ～ ～ ～

- 「要らないページ」の選び方のことを、
「**追出しアルゴリズム**」 または
「**置換えアルゴリズム**」 と呼ぶ

ページ追出し～置換えアルゴリズム

- ページの追出し方で、次に欲しいページが
主記憶上にあるか無いか（＝ページフォルトの
発生率）が変わる

ページ追出し～置換えアルゴリズム

- ページの追出し方で、次に欲しいページが主記憶上にあるか無いか（＝ページフォルトの発生率）が変わる
- 将来使うページが分かればそれを残せばいい（⇒ OPTアルゴリズム）

ページ追出し～置換えアルゴリズム

- ページの追出し方で、次に欲しいページが主記憶上にあるか無いか（＝ページフォルトの発生率）が変わる
- 将来使うページが分かればそれを残せばいい（⇒ OPTアルゴリズム）
 - ～ しかし、現実には未来はわからないので、実現できない

ページ追出し～置換えアルゴリズム

- ページの追出し方で、次に欲しいページが主記憶上にあるか無いか（＝ページフォルトの発生率）が変わる
- 将来使うページが分かればそれを残せばいい（⇒ OPTアルゴリズム）
 - ～ しかし、現実には未来はわからないので、実現できない
- 3つの例： FIFO、LRUと、理想的（OPT）

ページ追出し～置換えアルゴリズム

- 例として3つのアルゴリズムで追出しを比較しよう
- 条件として
 - アクセスするページの番号（参照列）が与えられている

ページ追出し～置換えアルゴリズム

- 例として3つのアルゴリズムで追出しを比較しよう
- 条件として
 - アクセスするページの番号（参照列）が与えられている
 - この順にアクセスしたときどう追出されるかを評価
 - 参照列の例（教p110～）

参照列

0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

ページ追出し～置換えアルゴリズム

- 例として3つのアルゴリズムで追出しを比較しよう
- 条件として
 - アクセスするページの番号（参照列）が与えられている
 - この順にアクセスしたときどう追出されるかを評価
 - 参照列の例（教p110～）

参照列

0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

– 物理ページの量

- この物理ページの量で、いつ空きがなくて追出すか

追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列

0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
				4	4	4	5		5	5	5			5
			1	0	0	0	0		0	0	0	3		3
		2	2	2	1	1	1	1	1	1	1	1	4	4
			3	3	7	2	2	2	2	2	2	2	2	2
			X	X	X	X	X					X	X	

物理メモリは
4ページある
とする

追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ0を 物理ページに 持ってきた 物理ページは 空いていた	0					4	4	4	5		5	5	5		5
					1	0	0	0	0		1	0	0	0	0
				2	2	2	1	1	1	1	1	1	1	4	4
				3	3	7	2	2	2	2	2	2	2	2	2
フォルト	X			X	X	X	X	X					X	X	

物理メモリは4ページあるとする

追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ1を 物理ページに 持ってきた 物理ページは 空いていた	0	0				4	4	4	5		5	5	5		5
		1				1	0	0	0		1	0	0	0	0
				2	2	2	1	1	1	1	1	1	1	4	4
				3	3	7	2	2	2	2	2	2	2	2	2
フォルト	X	X		X	X	X	X	X					X	X	

物理メモリは4ページあるとする

追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ2を 物理ページに 持ってきた 物理ページは 空いていた	0	0	0			4	4	4	5		5	5	5		5
		1	1			1	0	0	0		1	0			
			2			2	2	1	1		1	1	4	4	
						3	3	7	2		2	2	2	2	
フォルト	X	X	X	X	X	X	X	X	X				X	X	

物理メモリは4ページあるとする

追出し～F I F O（先入れ先出し）①

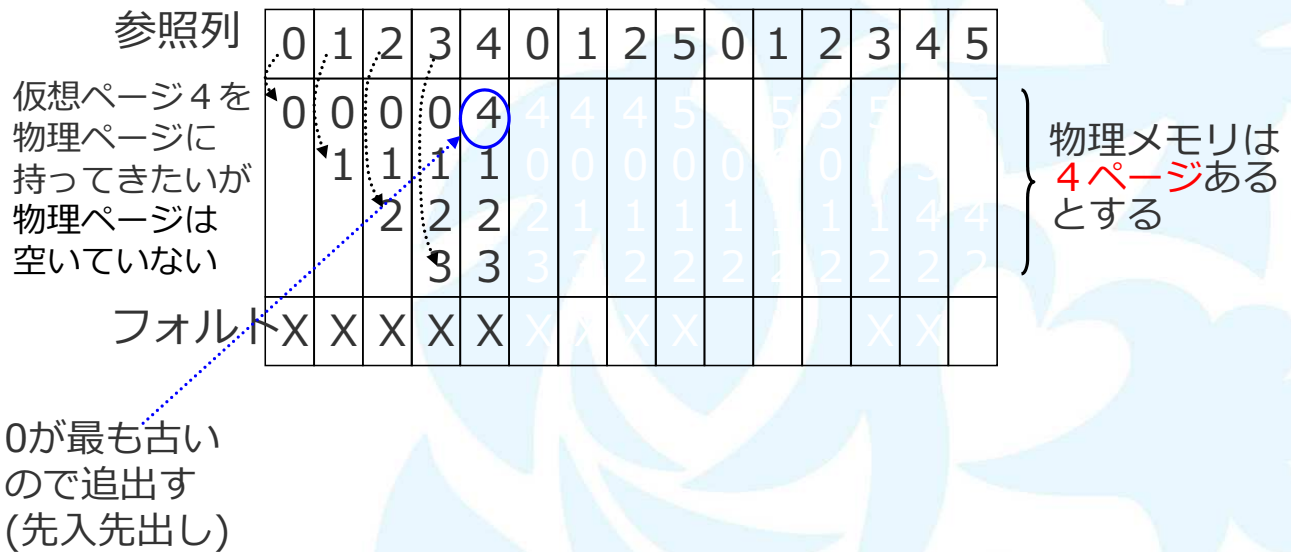
- 定義：最も古いページから先に追い出す

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ3を 物理ページに 持ってきた 物理ページは 空いていた	0	0	0	0		4	4	4	5		5	5	5		5
		1	1	1		1	0	0	0		1	0			
			2	2		2	2	1	1		1	1	4	4	
				3		3	3	7	2		2	2	2	2	
フォルト	X	X	X	X	X	X	X	X	X				X	X	

物理メモリは4ページあるとする

追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す



追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す



追出し～F I F O（先入れ先出し）①

- 定義： 最も古いページから先に追い出す

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
	0	0	0	0	4	4	4	4	5		5	5	5		
		1	1	1	1	0	0	0	0	0	0	0	0	0	0
			2	2	2	2	1	1	1	1	1	1	1	4	4
				3	3	3	2	2	2	2	2	2	2	2	2
フォルト	X	X	X	X	X	X	X	X					X	X	

物理メモリは
4ページある
とする

0が最も古い この状態では
ので追出す 1が最も古い
ので追出す

0を残して置けばよかったが
将来のことは分からない



追出し～F I F O（先入れ先出し）①

- 定義： 最も古いページから先に追い出す

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
	0	0	0	0	4	4	4	4	5		5	5	5		
		1	1	1	1	0	0	0	0	0	0	0	0	0	0
			2	2	2	2	1	1	1	1	1	1	1	4	4
				3	3	3	2	2	2	2	2	2	2	2	2
フォルト	X	X	X	X	X	X	X	X					X	X	

物理メモリは
4ページある
とする

0が最も古い この状態では この状態では
ので追出す 1が最も古い 2が最も古い
ので追出す ので追出す
(先入先出し)

0を残して置けばよかったが
将来のことは分からない



追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列

	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ2, 5 を物理ページに 持ってきたいが 物理ページは 空いていない	0	0	0	0	4	4	4	4	5						
		1	1	1	1	0	0	0	0						
			2	2	2	2	1	1	1						
				3	3	3	3	2	2						
フォルト	X	X	X	X	X	X	X	X					X	X	

物理メモリは4ページあるとする

0が最も古いので追出す この状態では1が最も古いので追出す この状態では2が最も古いので追出す

0を残して置けばよかったが将来のことは分からない

追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列

	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ0を 参照するが 既に物理ページに あるので新たに 持つてくる必要は ない	0	0	0	0	4	4	4	4	5	5					
		1	1	1	1	0	0	0	0	0					
			2	2	2	2	1	1	1	1					
				3	3	3	3	2	2	2					
フォルト	X	X	X	X	X	X	X	X					X	X	

物理メモリは4ページあるとする

0が最も古いので追出す この状態では1が最も古いので追出す この状態では2が最も古いので追出す 0があるので何も追出さない（そのまま使う）（フォルトなし）

0を残して置けばよかったが将来のことは分からない

追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列

	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ1を参照するが既に物理ページにあるので新たに持ってくる必要はない	0	0	0	0	4	4	4	4	5	5	5	5	5	5	5
フォルト	X	X	X	X	X	X	X	X					X	X	

物理メモリは4ページあるとする

0が最も古いので追出す
この状態では1が最も古いので追出す
この状態では2が最も古いので追出す
0があるので何も追出さない（そのまま使う）（フォルトなし）
1があるので何も追出さない（そのまま使う）（フォルトなし）

0を残して置けばよかったが将来のことは分からない



追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列

	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ2を参照するが既に物理ページにあるので新たに持ってくる必要はない	0	0	0	0	4	4	4	4	5	5	5	5	5	5	5
フォルト	X	X	X	X	X	X	X	X					X	X	

物理メモリは4ページあるとする

0が最も古いので追出す
この状態では1が最も古いので追出す
この状態では2が最も古いので追出す
0があるので何も追出さない（そのまま使う）（フォルトなし）
1があるので何も追出さない（そのまま使う）（フォルトなし）

0を残して置けばよかったが将来のことは分からない



追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列

	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ3, 4 を物理ページに 持ってきていたが 物理ページは 空いていない	0	0	0	0	4	4	4	4	5	5	5	5	5	5	5
		1	1	1	1	0	0	0	0	0	0	0	3	3	3
			2	2	2	2	1	1	1	1	1	1	1	1	4
				3	3	3	3	2	2	2	2	2	2	2	2
フォルト	X	X	X	X	X	X	X	X					X	X	

物理メモリは4ページあるとする

0が最も古いので追出す
この状態では1が最も古いので追出す
この状態では2が最も古いので追出す
0があるので何も追出さない（そのまま使う）（フォルトなし）
1があるので何も追出さない（そのまま使う）（フォルトなし）

0を残して置けばよかったが将来のことは分からない



追出し～F I F O（先入れ先出し）①

- 定義：最も古いページから先に追い出す

参照列

	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
仮想ページ5を 参照するが 既に物理ページに あるので新たに 持ってくる必要は ない	0	0	0	0	4	4	4	4	5	5	5	5	5	5	5
		1	1	1	1	0	0	0	0	0	0	0	3	3	3
			2	2	2	2	1	1	1	1	1	1	1	1	4
				3	3	3	3	2	2	2	2	2	2	2	2
フォルト	X	X	X	X	X	X	X	X					X	X	

物理メモリは4ページあるとする

0が最も古いので追出す
この状態では1が最も古いので追出す
この状態では2が最も古いので追出す
0があるので何も追出さない（そのまま使う）（フォルトなし）
1があるので何も追出さない（そのまま使う）（フォルトなし）

0を残して置けばよかったが将来のことは分からない



追出し～F I F O（先入れ先出し）①

- このようにして、
 - 参照列から次に参照する仮想ページ番号を取り出して
 - そのページが物理メモリ上にあるかを判定して
 - あればそのまま参照して終わり
 - 無ければ、
 - 空きページがあれば、そこへ持ってくる
 - 空きページが無ければ、FIFOのルールに従って追出すページを決めてそこへ新しいページを持ってくる

追出し～F I F O（先入れ先出し）②

- 物理メモリが5ページあったらどう変わるか

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
	0					0	0	0	5		5	5	5		1
				1	1	1	1	1	0	1	0	0	0		1
			2	2	2	2	2	2	2	2	1	1	1	1	1
			3	3	3	3	3	3	3	1	2	2	2	2	2
			4	4	4	4	4	4	4	4	4	3	3		
フォルト				X	X				X	X	X	X	X	X	

物理メモリは5ページあるとする

- 自分で埋めてみよ

追出し～F I F O（先入れ先出し）②

- 物理メモリが5ページあったらどう変わるか

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
	0	0	0	0	0	0	0	0	5	5	5	5	5	4	4
		1	1	1	1	1	1	1	0	0	0	0	0	0	5
			2	2	2	2	2	2	1	1	1	1	1	1	1
				3	3	3	3	3	3	3	2	2	2	2	2
					4	4	4	4	4	4	4	3	3	3	3
フォルト	X	X	X	X	X				X	X	X	X	X	X	X

物理メモリは5ページあるとする

追出し～F I F O（先入れ先出し）②

- 物理メモリが5ページあったらどう変わるか

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
	0	0	0	0	0	0	0	0	5	5	5	5	5	4	4
		1	1	1	1	1	1	1	0	0	0	0	0	0	5
			2	2	2	2	2	2	2	1	1	1	1	1	1
				3	3	3	3	3	3	3	3	2	2	2	2
					4	4	4	4	4	4	4	4	3	3	3
フォルト	X	X	X	X	X				X	X	X	X	X	X	X

物理メモリは5ページあるとする

ここら辺のフォルトは無くなったが

追出し～F I F O（先入れ先出し）②

- 物理メモリが5ページあったらどう変わるか

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
	0	0	0	0	0	0	0	0	5	5	5	5	5	4	4
		1	1	1	1	1	1	1	0	0	0	0	0	0	5
			2	2	2	2	2	2	1	1	1	1	1	1	1
				3	3	3	3	3	3	2	2	2	2	2	2
					4	4	4	4	4	4	3	3	3	3	3
フォルト	X	X	X	X	X				X	X	X	X	X	X	X

物理メモリは5ページあるとする

ここら辺のフォルトは無くなったが
後ろの方ではむしろフォルトが増えている



追出し～F I F O（先入れ先出し）②

- 物理メモリが5ページあったらどう変わるか

参照列	0	1	2	3	4	0	1	2	5	0	1	2	3	4	5
	0	0	0	0	0	0	0	0	5	5	5	5	5	4	4
		1	1	1	1	1	1	1	0	0	0	0	0	0	5
			2	2	2	2	2	2	2	1	1	1	1	1	1
				3	3	3	3	3	3	3	2	2	2	2	2
					4	4	4	4	4	4	4	3	3	3	3
フォルト	X	X	X	X	X				X	X	X	X	X	X	X

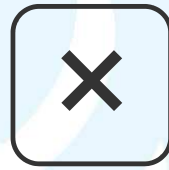
物理メモリは5ページあるとする

Beladyの異常（FIFOの異常）

物理メモリを増やしたのにフォルト率がかえって増えることがある



F I F O（先入れ先出し）の考え方が
理解できましたか？



↓
次へ