

オペレーティングシステム

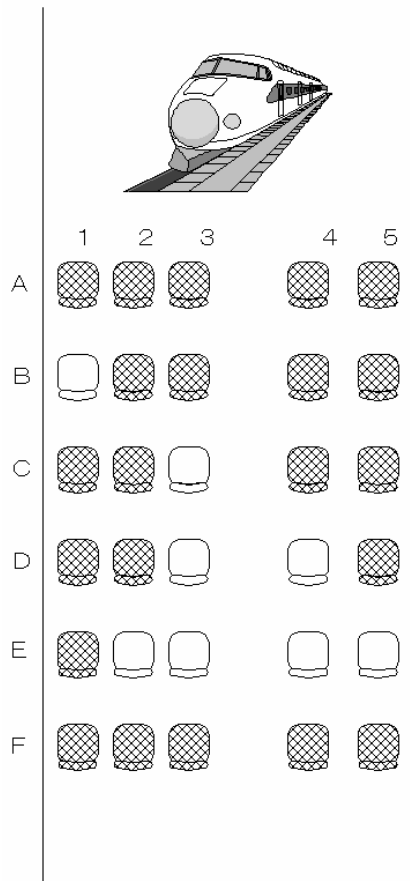
第3回(2009.04.30)

デッドロックと排他制御

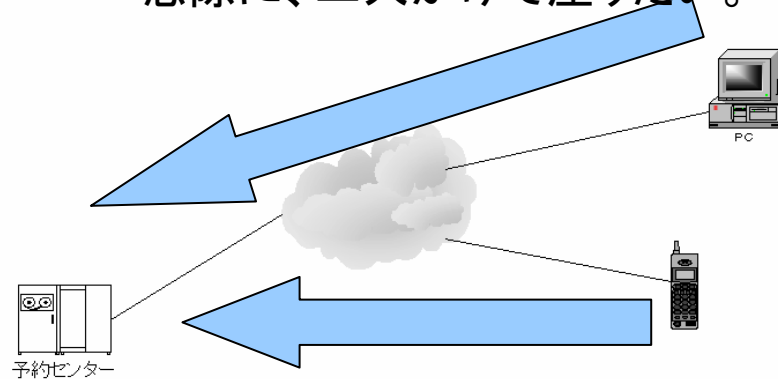
排他制御とは？

- 誰かが使っている時に、他の人に使わせない、他の人に使用させない。
- 座席予約
 - 電話をかけた。
 - 残り数枚になりました。
 - インターネットや、携帯電話、窓口などで、複数の予約が入った

座席予約の場合



窓際に、二人がけで座りたい。



空き状態: B1, C3, D3, D4, E2-E5

窓際に、二人がけで座りたい。

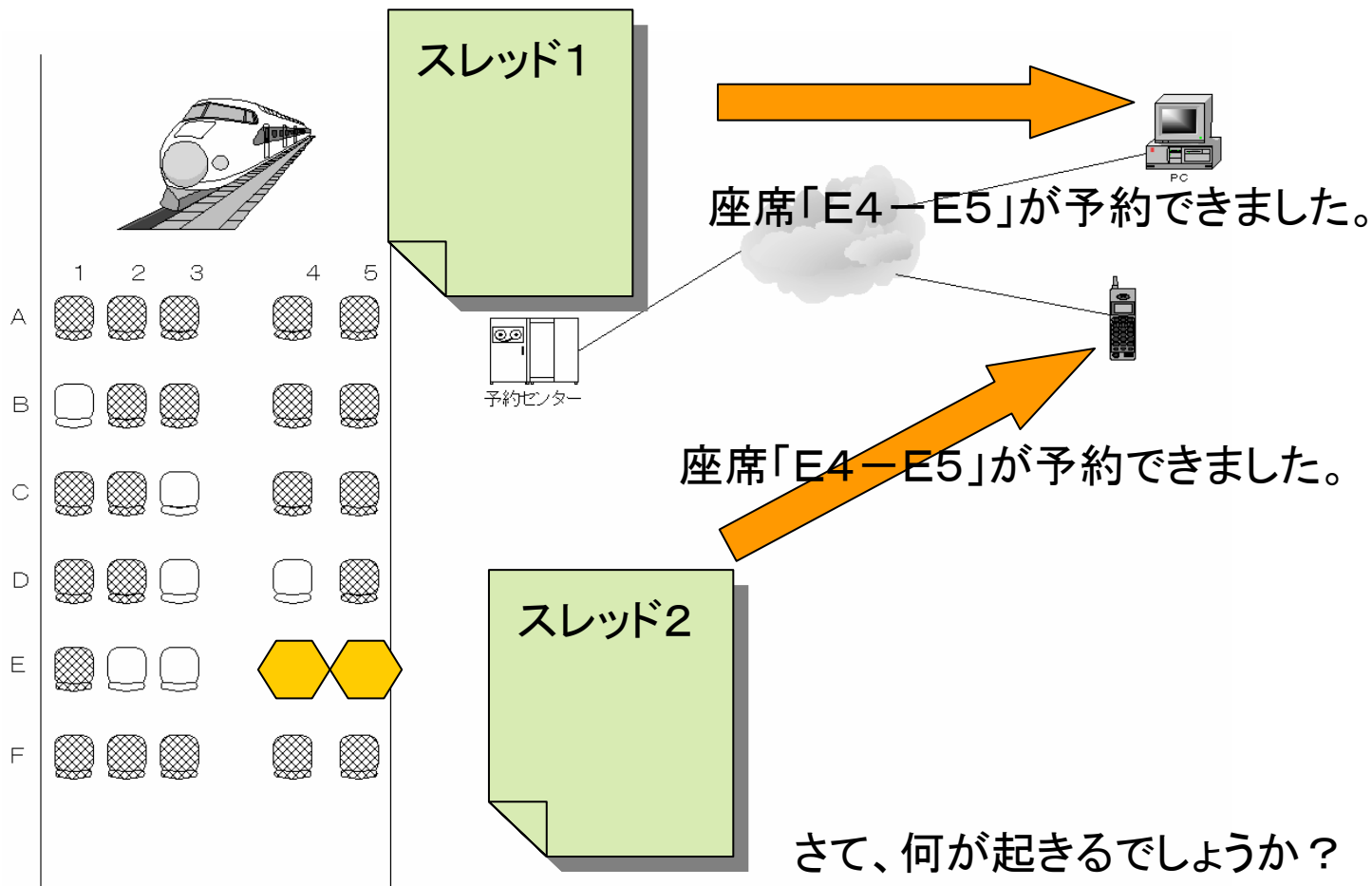
スレッド1
起動

スレッド2
起動

各スレッドの論理

- 「窓際に二人がけ」で該当する座席を探す。
- ほとんど同時にアクセスがあったため、どちらのスレッドも同じ「空き席状況」の情報を持っている。
- 「スレッドって何ですか？」

座席予約の場合



スレッドって何ですか？

- 一つのホームページ（WEBページ）に複数の人がアクセスしています。
- 誰もが同じ画面を見ている、「入力」が違えば返す画面が違う。
 - 一つのプログラムが、複数同時に「ユーザ対応」していて、この一つ一つの「対応窓口」をスレッドと考えてください。
 - 「特定のセッションIDに結び付けられた、エンティティ」
- 詳しくは、第5回「割込みプロセス制御」で
- 教科書P95

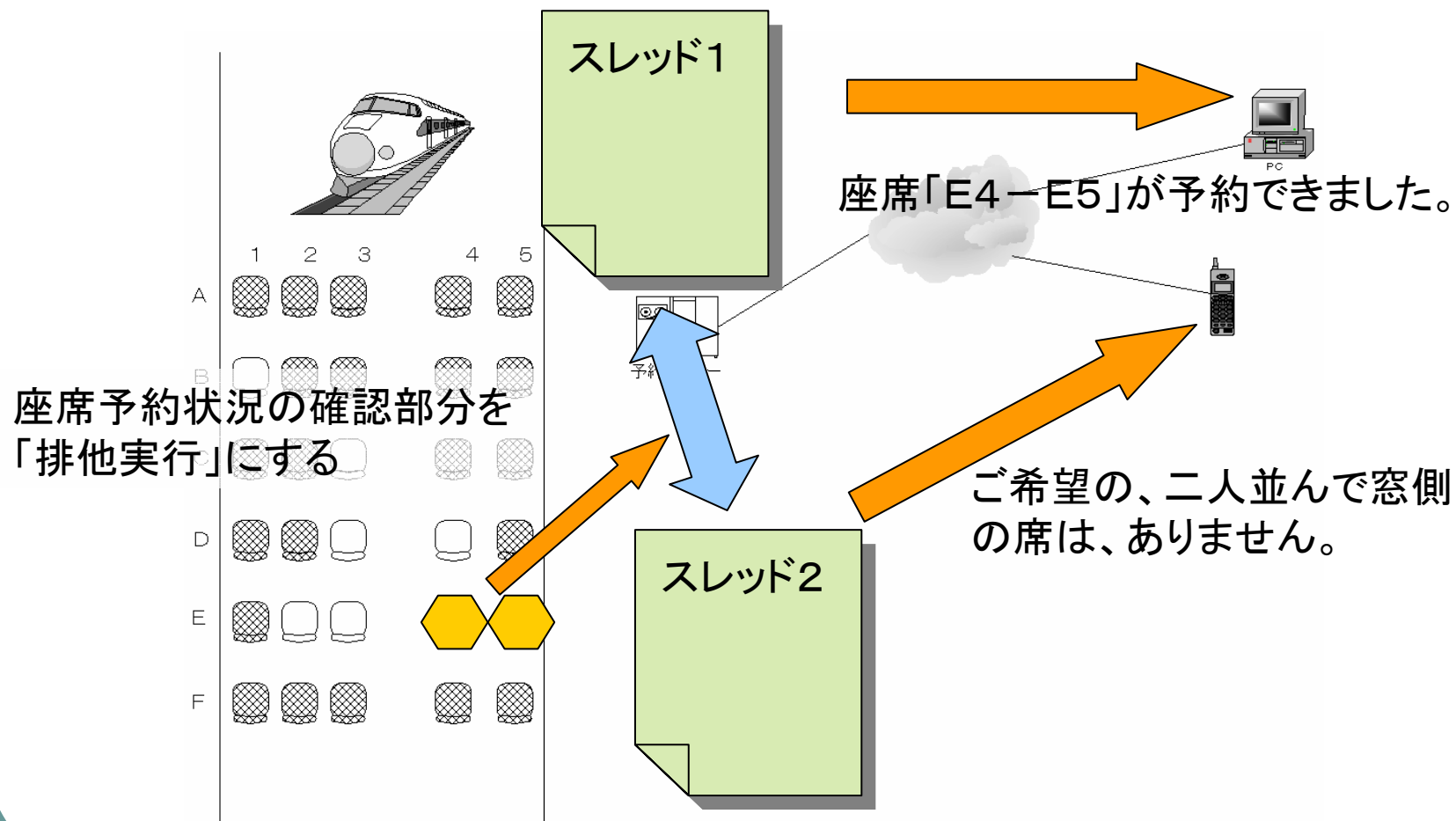
エンティティって何ですか？

- 「実体」のことです。
 - 「プログラム」は、「機械語が記述されたファイル」としてコンピュータに存在する。
 - 「プログラム」が起動されると「プロセス」として
 - メモリ
 - 入出力端末(ポート)
 - 実行状態管理メモリ
 - などを割り付けられる。
 - 詳しくは、第5回「割込み・プロセス制御」で
 - 教科書P83
 - この「プログラム、メモリ、入出力」などを割り付けられて、今実行しているプロセスや、そのプロセスが使用している資源全体を「エンティティ」と呼ぶ。

排他制御

- 教科書P103
- 二つのスレッドが協調して動作する状況が発生した。

座席予約の場合



アクセスの競合

- 教科書P105
- 同一メモリ(記憶域)に競合してアクセスすると、結果不正が起こる。
- それぞれの処理が「排他的」に「順次に」しか実行できないようにする。
- このような実行部分を「クリティカルセクション」と呼ぶ。

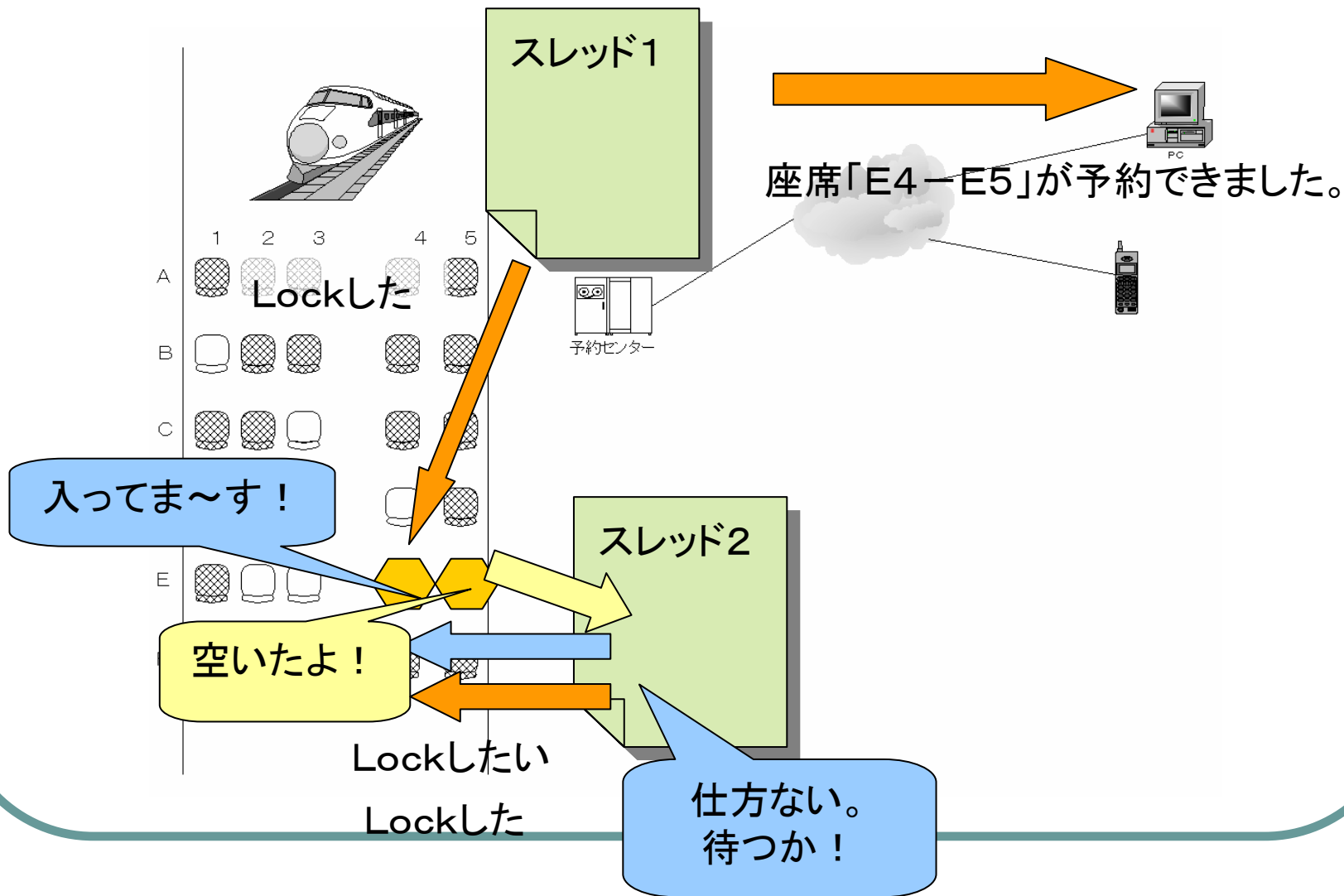
LockとUnlock

- クリティカルな資源（排他実行したい資源）にロックをかける。
- 教科書P107

クリティカルって？

- 「一番重要！」
- クリティカルパス
 - 文化祭で、「たこ焼き屋」をやりました。
 - フロアは30席、ウェイターは5人、調理当番は3人準備した。
 - ところが、「たこ焼き器」は、家庭用が一つしかない。
 - この場合、「たこ焼き器」がクリティカルパス
 - ウェイターは5人、たこ焼き器は10台、調理当番5人を準備した。
 - ところが、フロアには、座席が一つしかない。
 - この場合、「フロアの座席」がクリティカルパス
- 最も少ない資源に「要求」が集中するような部分のこと

座席予約の場合



セマフォ(Semaphore)

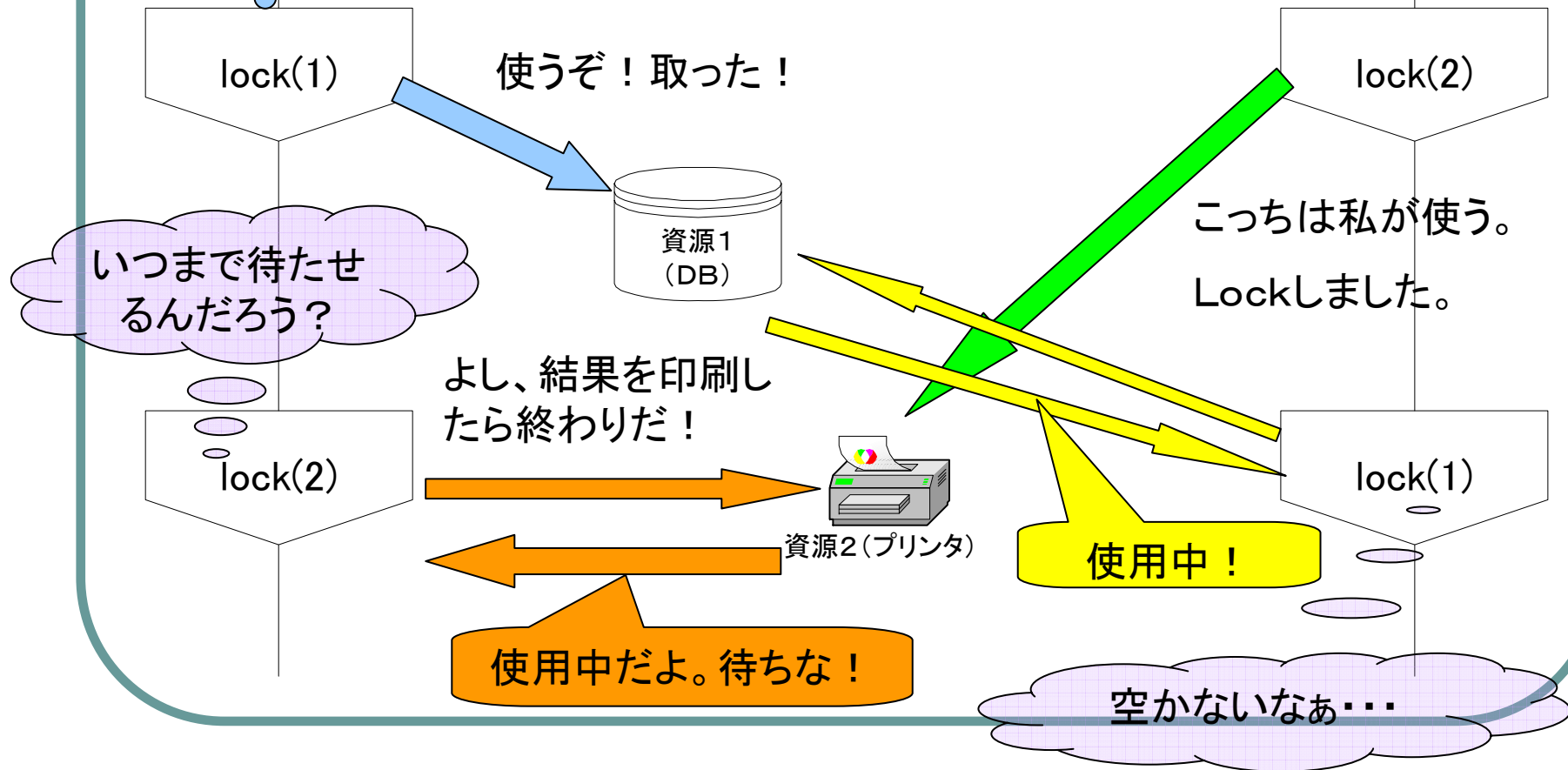
- ロックとアンロックの場合、「二つのプロセス」以上には対応しきれない。
- セマフォを用いると、複数のプロセスの競合状況をうまくプログラムできる。

デッドロック

- 教科書P114
- 同期の取り方がまずくて、処理が先に進めなくなってしまった状態

デッドロックの例

かくして、二つのスレッドは、「相手が資源を手放す」のを永遠に待ち続けたのでした…。めでたし、めでたし??



まとめ

- 排他制御とは、「データの整合性」を確保するために、独占的に資源にアクセスする、など、「競合」が起きないようにするために行われる。
- スレッドやプロセスの間で連携を取る必要がある。
- 排他制御には、LockとUnLockという方法がある。
- 多数のプロセスが排他制御を行うために、セマフォというメカニズムを使用する場合がある。
- 排他制御に失敗すると、デッドロックが発生する。