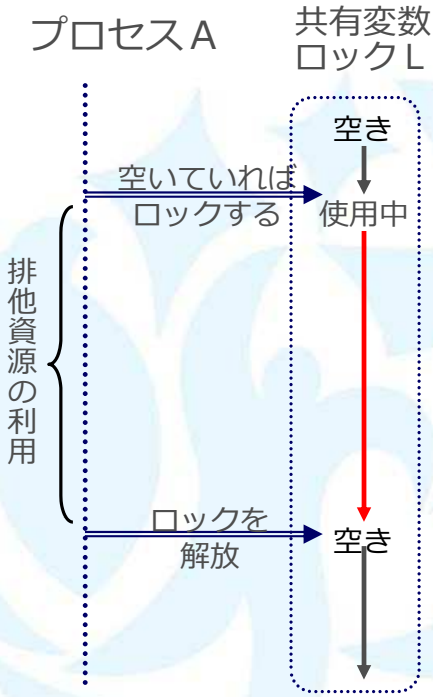
排他制御の仕組2

ビジーウェイトとその問題

- ビジーウェイト
  ⇒
  ロックフラッグを利用して相互排他を実現する方法の１つ

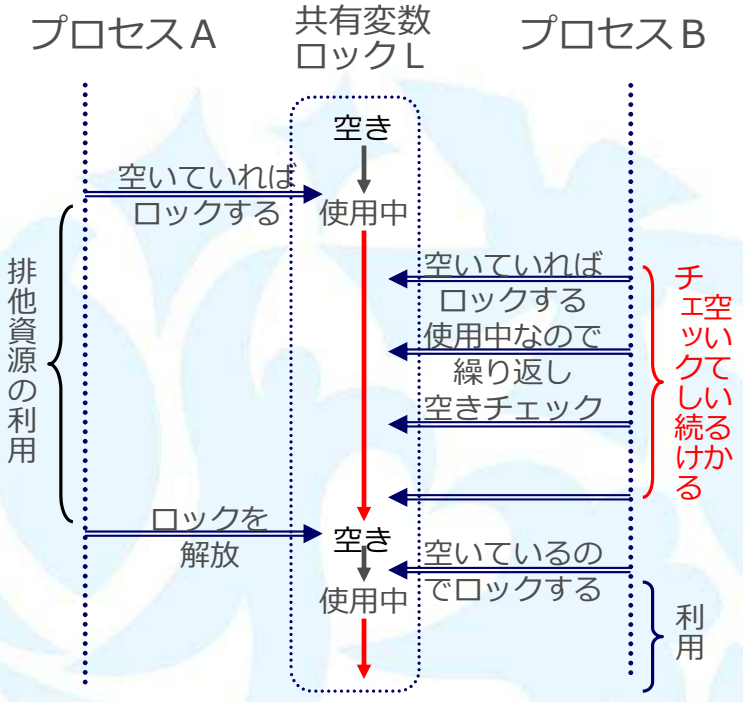
ビジーウェイトとその問題

- ビジーウェイト
  ⇒
  ロックフラッグを利用して相互排他を実現する方法の１つ
- 使用中 (ビジー) なら空くまで、繰り返し使用中かを尋ねる



ビジーウェイトとその問題

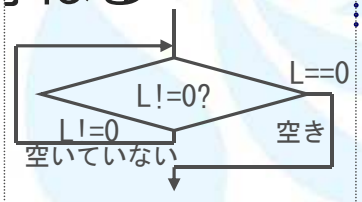
- ビジーウェイト
  ⇒
  ロックフラッグを利用して相互排他を実現する方法の１つ
- 使用中 (ビジー) なら空くまで、繰り返し使用中かを尋ねる

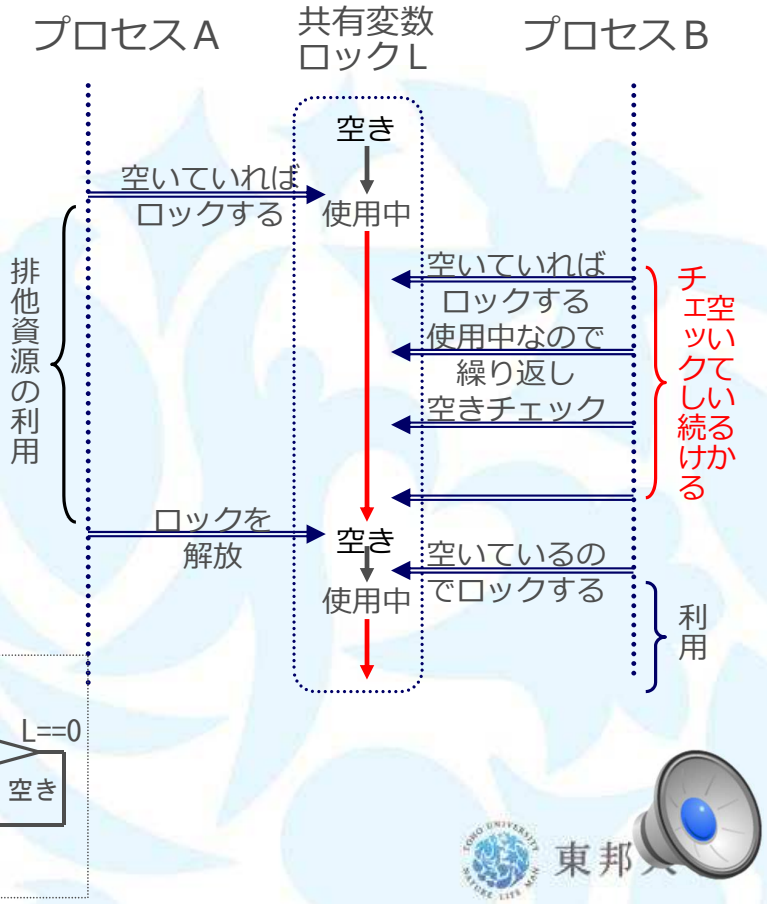


ビジーウェイトとその問題

- ビジーウェイト
  ⇒
  ロックフラッグを利用して相互排他を実現する方法の１つ
- 使用中 (ビジー) なら
  空くまで、繰り返し
  使用中かを尋ねる

```
        :
while (L!=0)
   { 何もしない }
ロック取れたので進む
        :
```

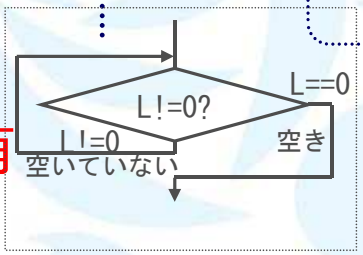




ビジーウェイトとその問題

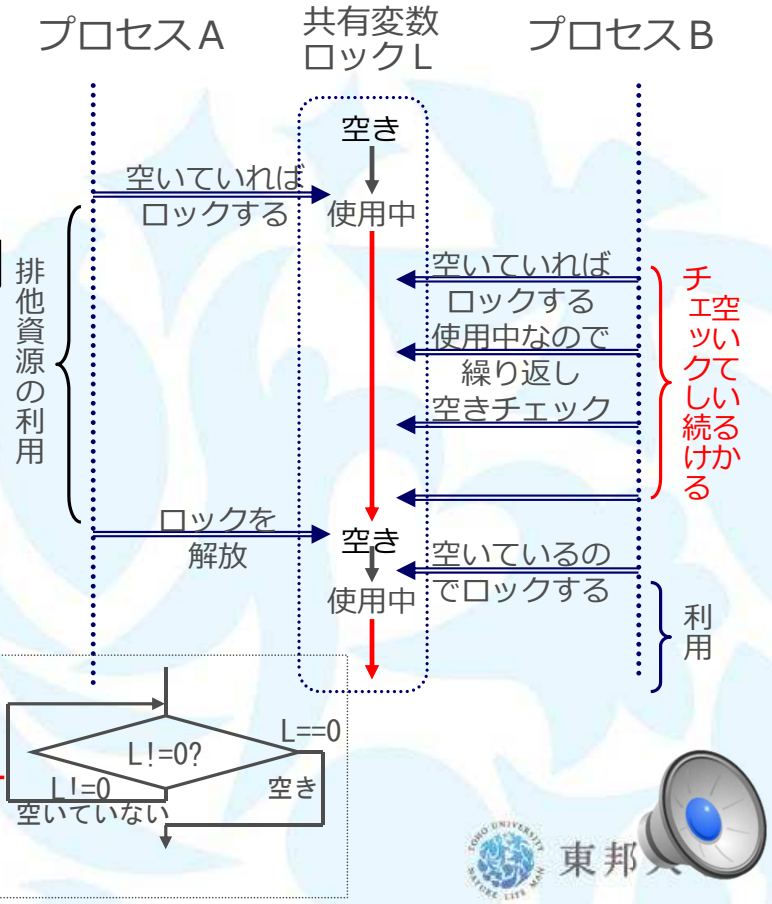
- ビジーウェイト
  ⇒
  ロックフラッグを利用して相互排他を実現する方法の１つ
- 使用中 (ビジー) なら
  空くまで、繰り返し
  使用中かを尋ねる

- 待っている間は
  何もしないでCPU占有



ビジーウェイトとその問題

- ビジーウェイト
  ⇒
  ロックフラッグを利用して相互排他を実現する方法の１つ
- 使用中 (ビジー) なら空くまで、繰り返し使用中かを尋ねる
- 待っている間は何もしないでCPU占有
  ⇒ CPUの無駄遣い

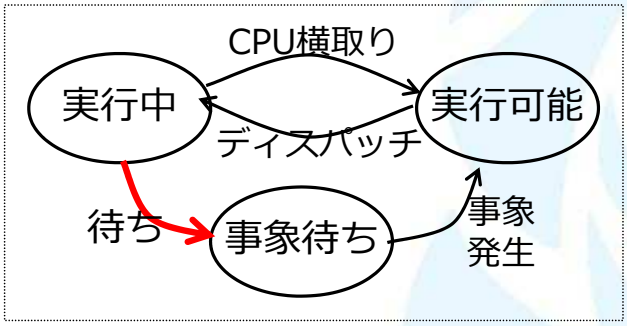


プロセスの状態遷移を使った待ち

- 待っている間何もしないでCPU占有 ⇒ 無駄

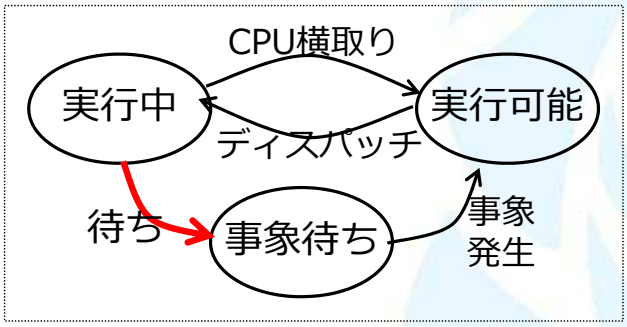
プロセスの状態遷移を使った待ち

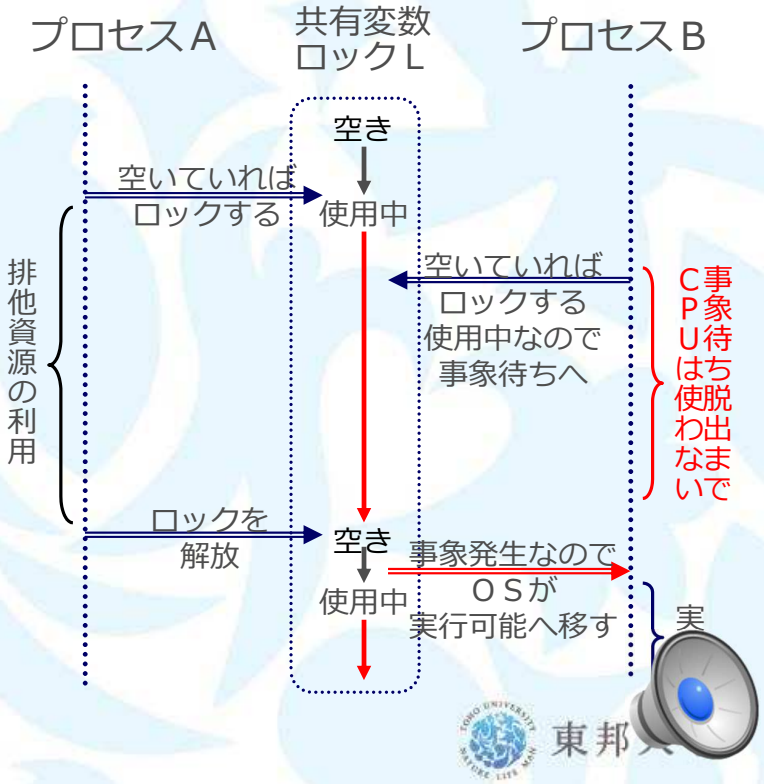
- 待っている間何もしないでCPU占有 ⇒ 無駄
  ↓
- なら、CPUから追出せ
  ⇒ 「事象待ち」へ



プロセスの状態遷移を使った待ち

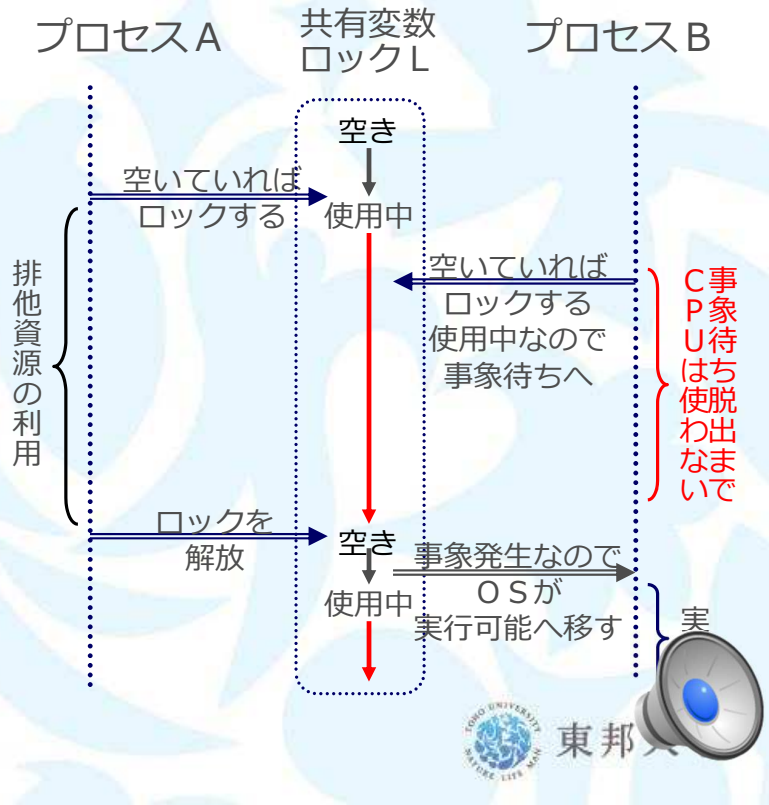
- 待っている間何もしないでCPU占有 ⇒ 無駄
  ↓
- なら、CPUから追出せ
  ⇒ 「事象待ち」へ





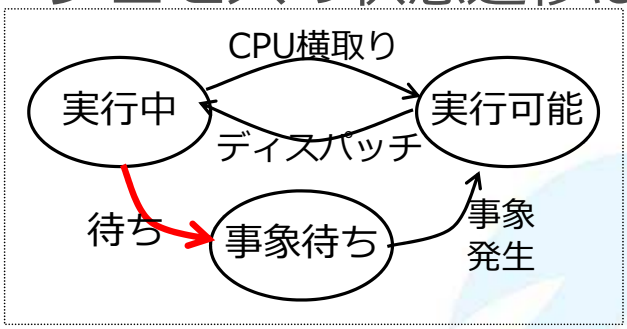
プロセスの状態遷移を使った待ちの問題

- プロセスの状態遷移は

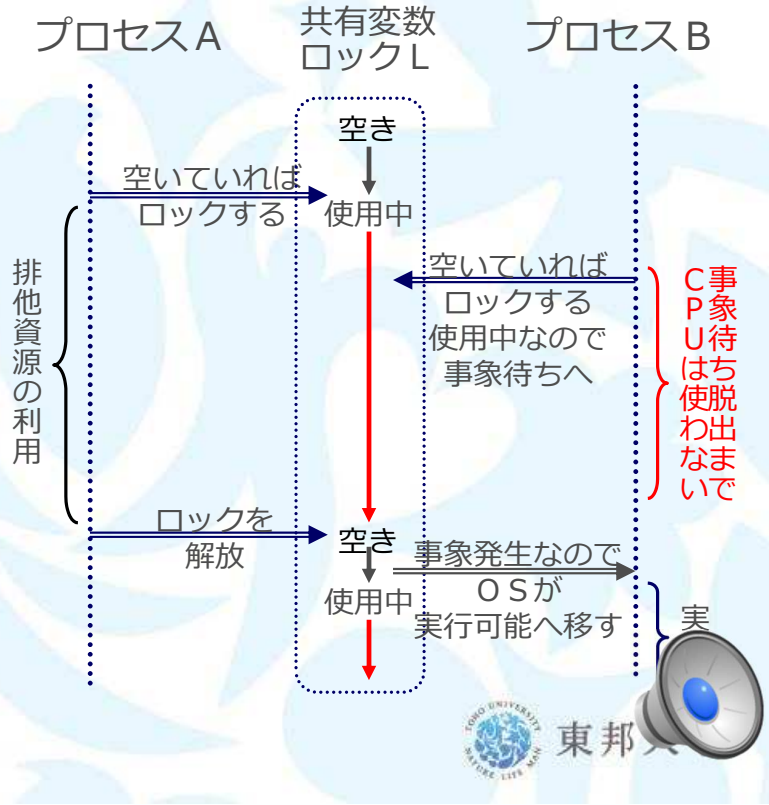


プロセスの状態遷移を使った待ちの問題

- プロセスの状態遷移は

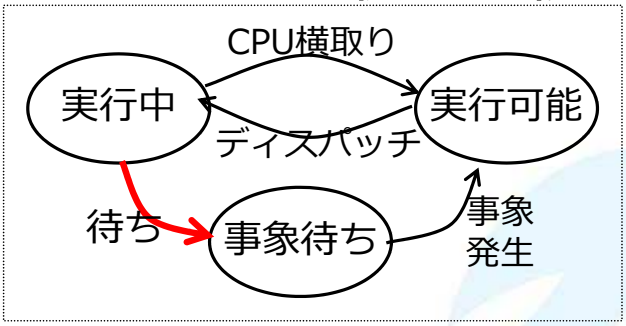


遷移に時間がかかる



プロセスの状態遷移を使った待ちの問題

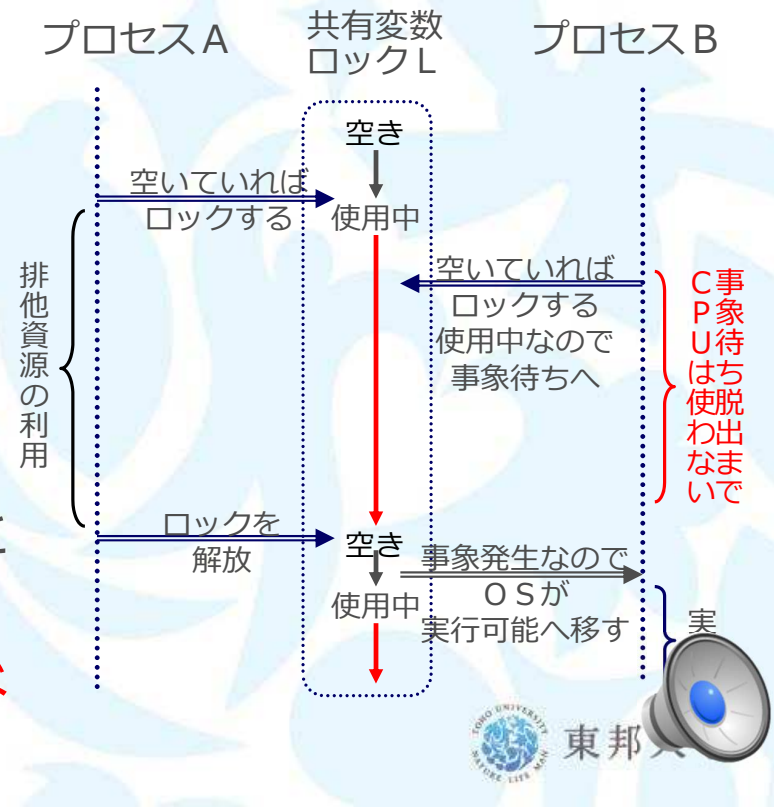
- プロセスの状態遷移は



遷移に時間がかかる

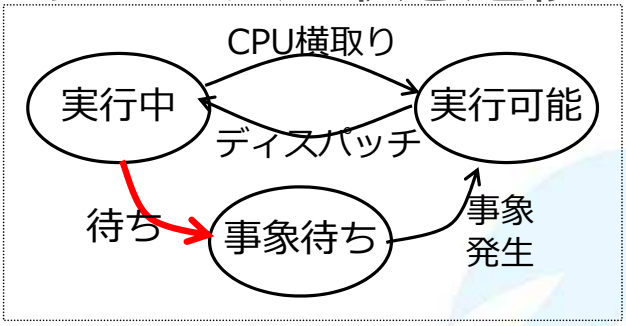
↓

- 排他中の時間が短いと
  状態遷移の時間が
  大きなオーバーヘッド



ハイレベルの待ち

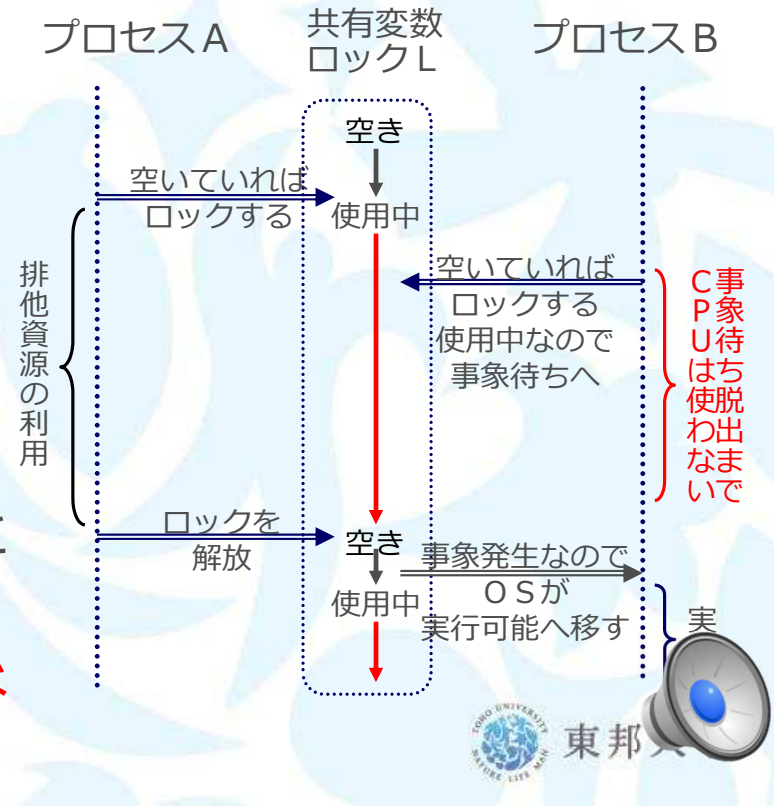
- プロセスの状態遷移は



遷移に時間がかかる

↓

- 排他中の時間が短いと
  状態遷移の時間が
  大きなオーバーヘッド



排他制御の仕組のまとめ

- 排他の仕組としては

東邦大

排他制御の仕組のまとめ

- 排他の仕組としては
  共有変数「ロック中」を使ってビジーウェイト
  することができるが

東邦大

排他制御の仕組のまとめ

- 排他の仕組としては
  共有変数「ロック中」を使ってビジーウェイト
  することができるが
    待っている間 CPUを無駄遣いする

排他制御の仕組のまとめ

- 排他の仕組としては
  共有変数「ロック中」を使ってビジーウェイト
  することができるが
    待っている間 CPUを無駄遣いする
- もう１つの仕組として、

排他制御の仕組のまとめ

- 排他の仕組としては
  共有変数「ロック中」を使ってビジーウェイト
  することができるが
    待っている間 CPUを無駄遣いする
- もう１つの仕組として、
  プロセスの状態を「事象待ち」に遷移させる

排他制御の仕組のまとめ

- 排他の仕組としては
  共有変数「ロック中」を使ってビジーウェイト
  することができるが
    待っている間 CPUを無駄遣いする
- もう１つの仕組として、
  プロセスの状態を「事象待ち」に遷移させる
  （ロック待ちが終わったら実行可能に戻す）

排他制御の仕組のまとめ

- 排他の仕組としては
  共有変数「ロック中」を使ってビジーウェイト
  することができるが
    待っている間 CPUを無駄遣いする
- もう１つの仕組として、
  プロセスの状態を「事象待ち」に遷移させる
  （ロック待ちが終わったら実行可能に戻す）
    この方法ならCPUの無駄遣いは起こらない

排他制御の仕組のまとめ

- 排他の仕組としては
  共有変数「ロック中」を使ってビジーウェイト
  することができるが
    待っている間 CPUを無駄遣いする
- もう１つの仕組として、
  プロセスの状態を「事象待ち」に遷移させる
  （ロック待ちが終わったら実行可能に戻す）
    この方法ならCPUの無駄遣いは起こらない
    しかし、状態遷移に時間がかかる

排他制御の仕組のまとめ

- 排他の仕組としては
  共有変数「ロック中」を使ってビジーウェイトすることができるが
  待っている間 CPUを無駄遣いする
- もう１つの仕組として、
  プロセスの状態を「事象待ち」に遷移させる
  （ロック待ちが終わったら実行可能に戻す）
  この方法ならCPUの無駄遣いは起こらない
  しかし、状態遷移に時間がかかる
  ロック中期間が短いとオーバーヘッド過

排他制御の仕組のまとめ

- 排他の仕組としては
  共有変数「ロック中」を使ってビジーウェイトすることができるが
  待っている間 CPUを無駄遣いする
- もう１つの仕組として、
  プロセスの状態を「事象待ち」に遷移させる
  （ロック待ちが終わったら実行可能に戻す）
  この方法ならCPUの無駄遣いは起こらない
  しかし、状態遷移に時間がかかる
  ロック中期間が短いとオーバーヘッド過

- どちらを選ぶかはロック中時間で決める

排他制御の仕組が理解できましたか？

○

次へ

×