

分散システム基礎と クラウドでの活用

レポート課題

国立情報学研究所

石川 冬樹

f-ishikawa@nii.ac.jp



課題：相互排他制御

■ シナリオ

- 複数のプロセス(固定)が動作しており, それらは時折, ある資源を利用し作業する
- 排他制御: 資源の利用が, 複数のプロセスにより同時になされることが決してないようにしたい
 - 例: ディスク領域を読み書きして処理している間に, 他のプロセスの書き込みが混ざること避ける
- 公平性: あるプロセスが「ずっと」その資源を利用できないことがないようにする
 - 永遠に利用できなかったり, 他プロセスと比べ偏った(不公平な)理由で長い時間利用できなかったり



課題：相互排他制御

- そのような制御の実現方法(プロトコル)を以下の2つの方針それぞれで検討する
 - 資源の利用可否の判断をする調整者を設け, 他プロセスは調整者に対して問い合わせや報告をして進めていく(もしくは, 調整者から各プロセスに連絡するかもしれない)
 - 資源の利用権をプロセス間である順序で受け渡していく



課題：相互排他制御

1. それぞれプロトコル(アルゴリズム)を記述せよ
実装詳細は捨象し抽象的に記述してよいが、下記の点を明確に説明できるようにする

- 排他制御が実現される理由
- 公平性の解釈とそれが実現される理由
- 起きうる障害の種類とタイミングの列挙(重要度で絞ってもよい)と、それらに対する対応方法

(仮定が必要なら、適宜導入する)



課題：相互排他制御

2. 記述した2つのプロトコルを比較，議論せよ

比較の観点は下記に挙げておくが，それぞれ「重要でない」と思えばその旨明記すればよく，また観点を追加してもよい

- 交換するメッセージ数のコスト
- 資源が必要となってから，利用可能となるまでの待ち時間
- 機能が維持できなくなったり性能が大きく悪化したりする障害の有無や種類，その可能性の高さ
- 実装やテスト，運用監視・管理などの行いやすさ