

コンピュータシステムB -ソフトウェアを中心に -

#5 マルチプログラミング

Yutaka Yasuda

システムソフトウェアの価値

- システムソフトウェアは何のためにあるか？
 - ハードウェア資源の利用効率を上げるため
 - 一度使った資源を迅速に、効率よく再利用したい
- なぜそれが必要だったのか？
 - 高価だったから
 - 性能が低かったから
- 「限られたハードウェア能力を限界まで使いたい」

ハードウェアの性能向上

- ハードウェアの高性能化に伴い

より多くの資源を積みながら、かつより高効率に利用する方向へ進化

- マルチプログラミング

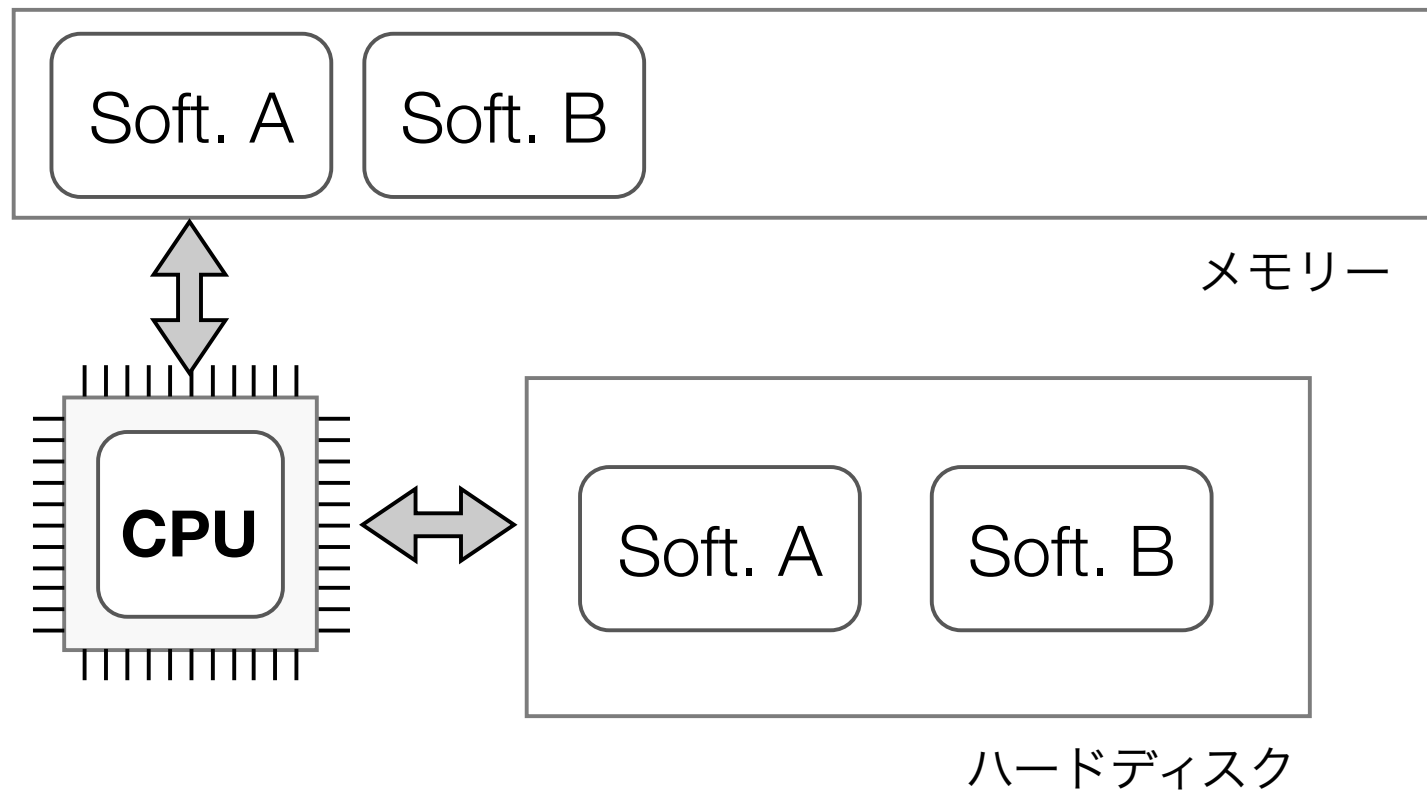
複数のソフトウェアを並行に実行したい

- CPU 処理能力やメモリ量などの向上による

向上した資源をよりよく使うためでもある

ニワトリタマゴな状態

マルチプログラミング



メインフレーム

- 大型・高価・少数利用者
- 要求に対して低い性能
- 利用形態

占有的に利用者に時間貸し

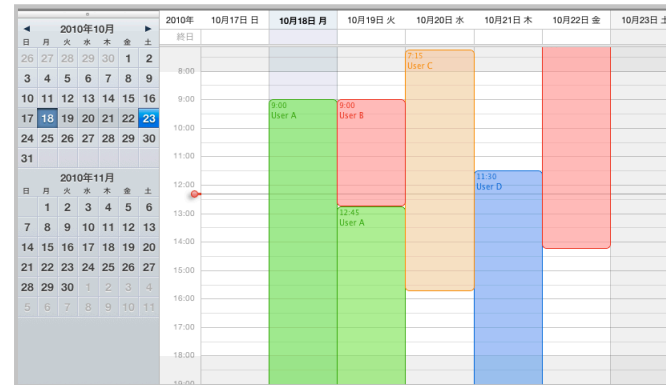
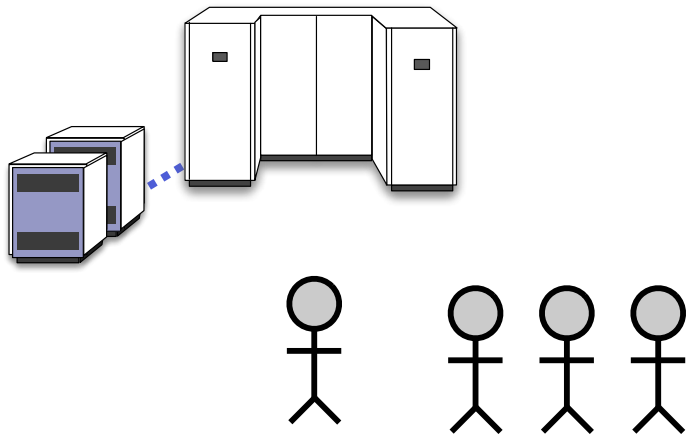
オペレータが順ぐりに仕事を掛ける

- ハードウェアの「利用効率最大化」が目的

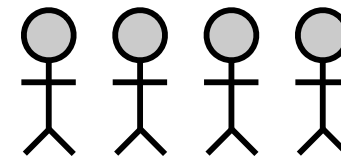
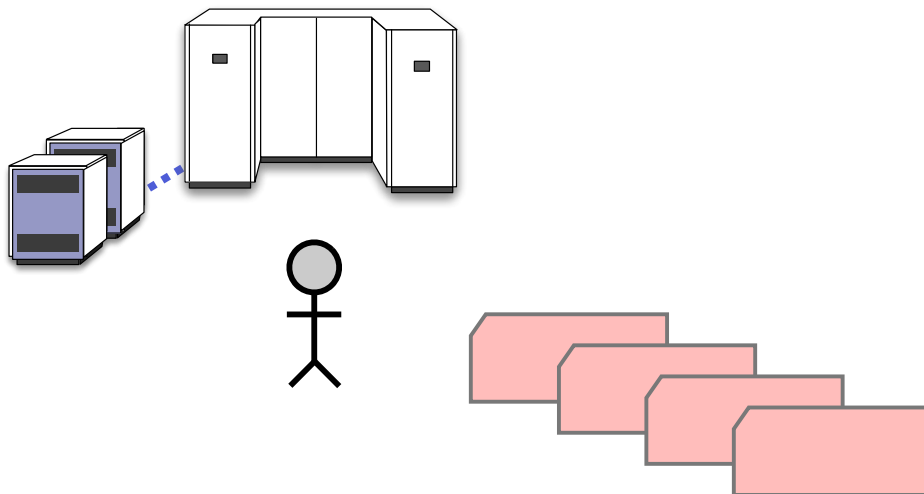


TOSBAC 3400 (1964)

時間貸し vs バッチ (一括) 処理依頼

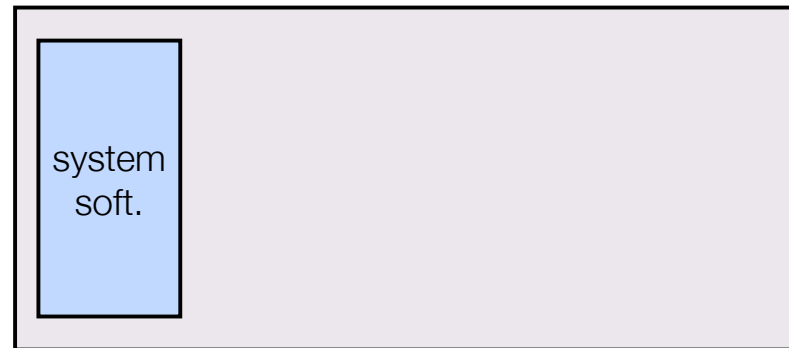


スケジュールを組んで人が並ぶ

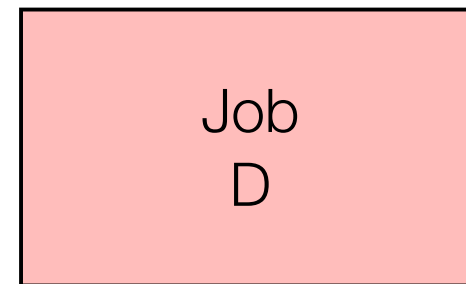
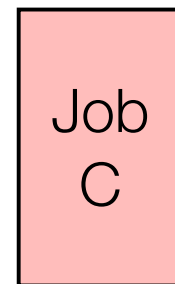
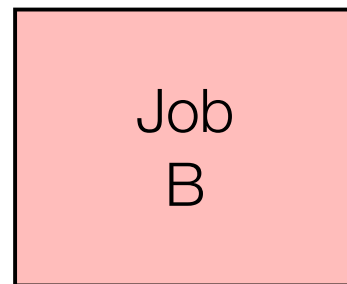
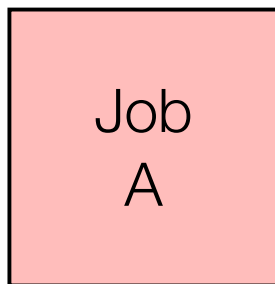


オペレータに預けて Job が並ぶ
(終わったら出力を受け取る)

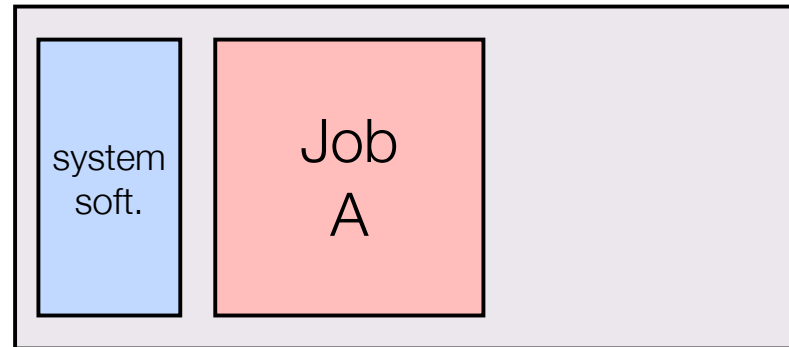
バッチ処理：メモリの状況



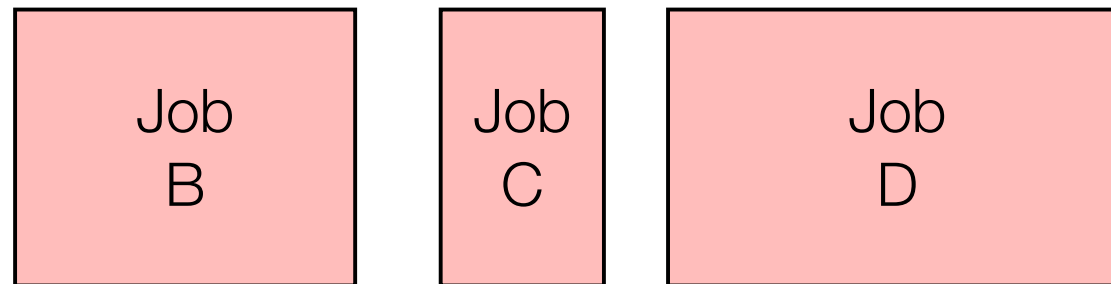
A, B, C, D の順に仕事を処理する



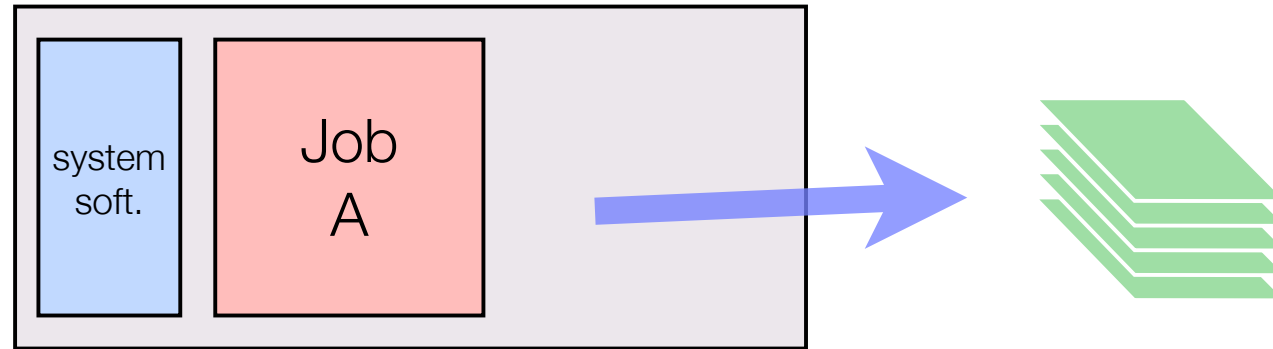
バッチ処理：メモリの状況



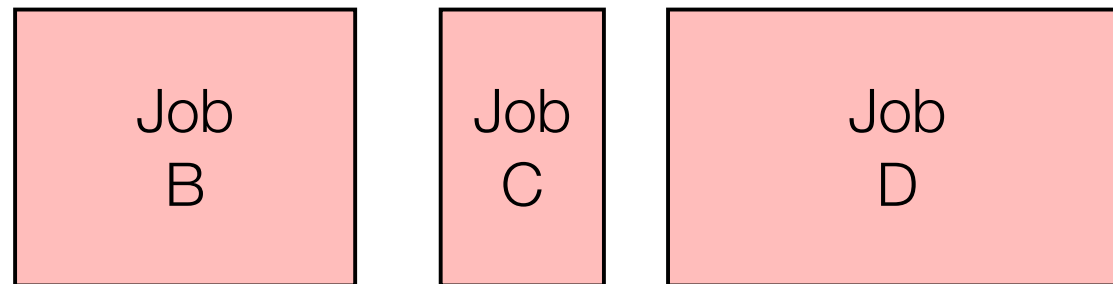
Aをメモリに読み込んで処理



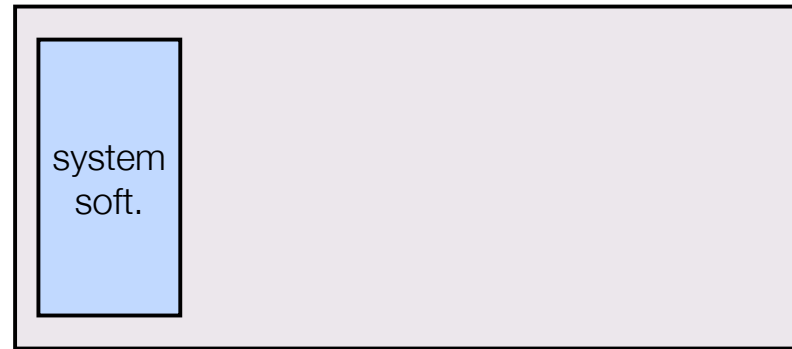
バッチ処理：メモリの状況



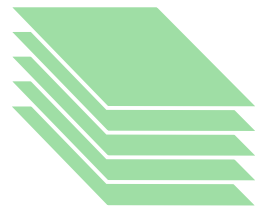
何らかの出力を行い



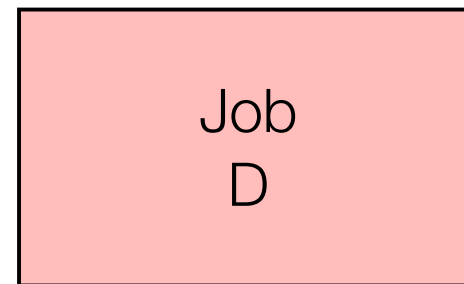
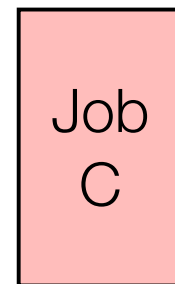
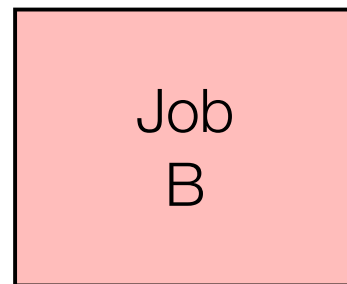
バッチ処理：メモリの状況



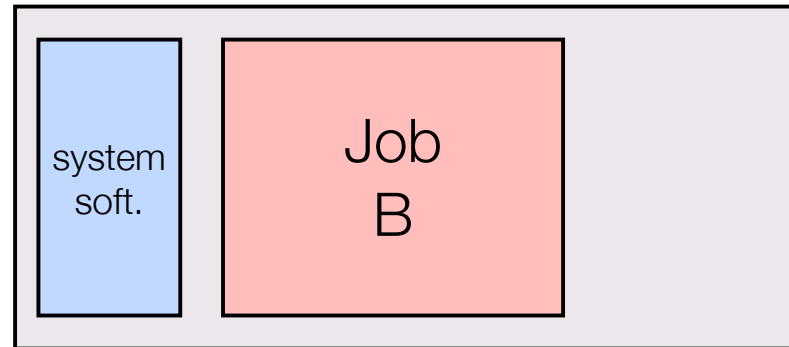
メモリから消滅



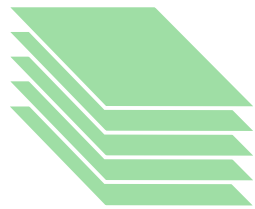
output A



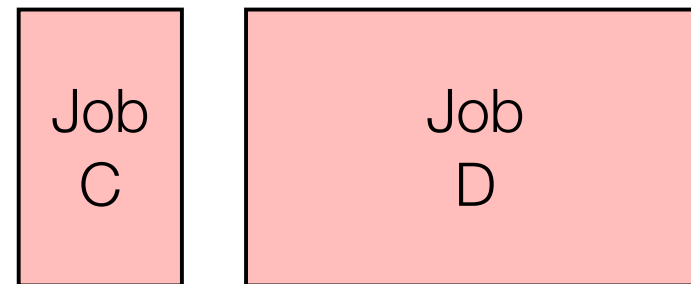
バッチ処理：メモリの状況



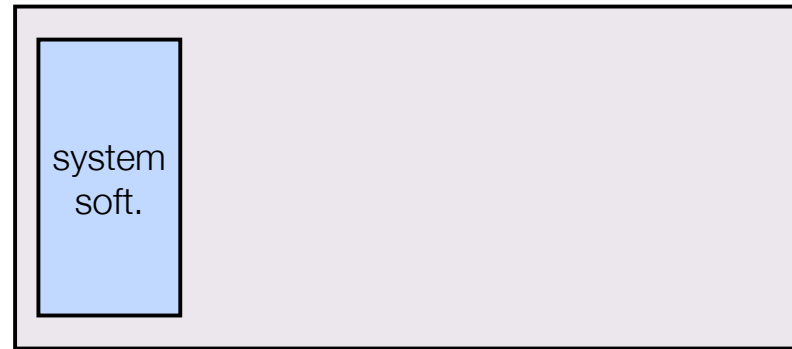
次はBを実行



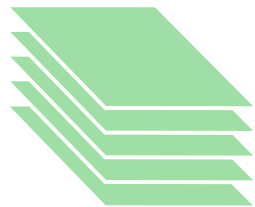
output A



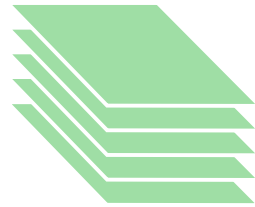
バッチ処理：メモリの状況



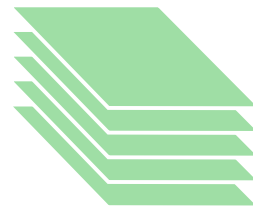
全ての Job の実行を完了（初期状態に戻る）



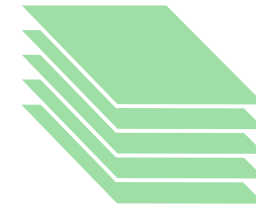
output A



output B



output C



output D

バッチ（一括）処理

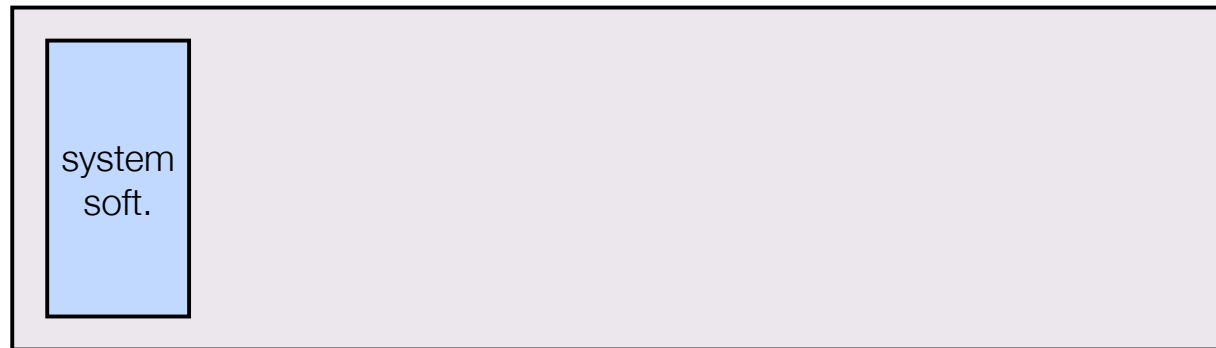
- 単一の処理プログラムを順繰りに実行する

当時はオペレータに依頼

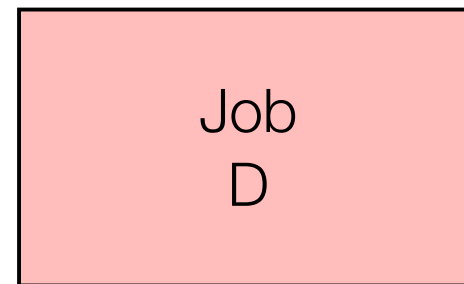
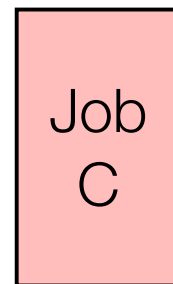
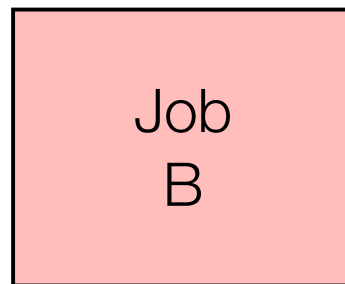
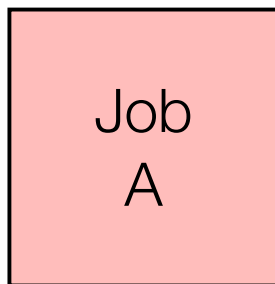
- 単一のプログラムでメモリもCPUも使い切るような状況では有効

現在でもスパコンの運用はこれに類する

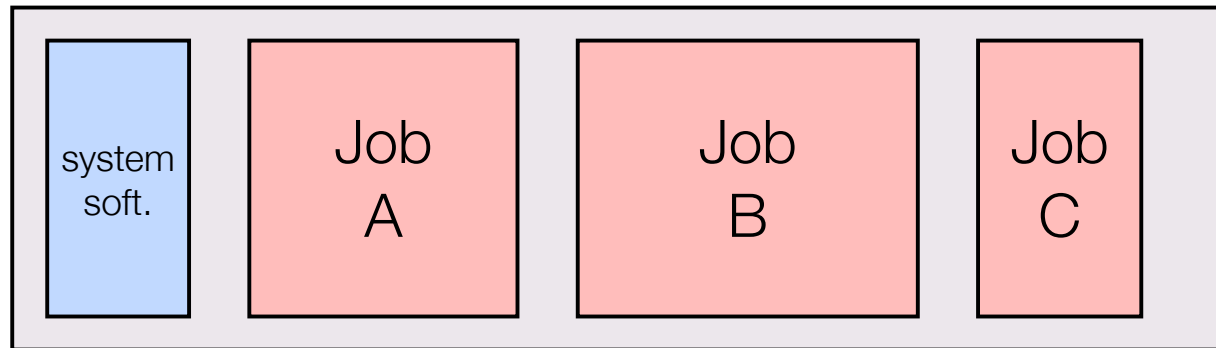
TSS (Time Sharing System)



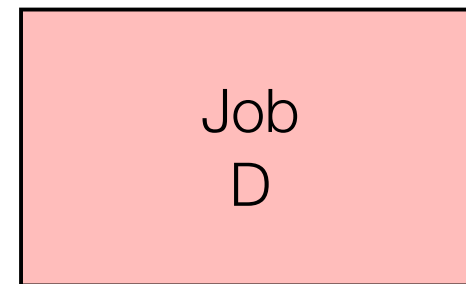
搭載メモリの増大



TSS (Time Sharing System)



複数のプログラムがメモリに載る

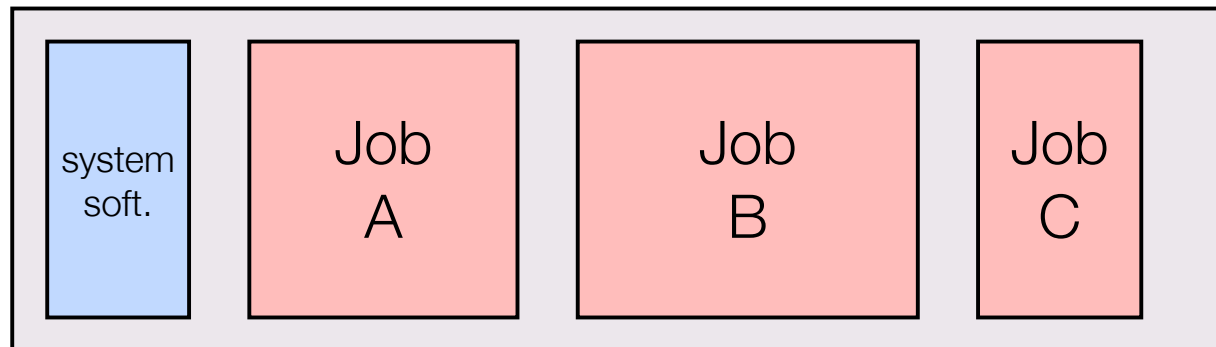


TSS (Time Sharing System)

- 並行処理とは言うが実際には順繰り処理

1つの CPU がある時間に実行できるのは1つの処理だけ

- Job A を実行して、何かをきっかけに Job B の処理に移る
しばらくすると一周する



TSS (Time Sharing System)

- 複数のプログラムがメモリに載る
- CPU の遊休時間ができる (I/O待ちやオペレータ待ち)
- マルチプログラミング：複数のプログラムを並行に処理

プロセス：実行中のプログラム

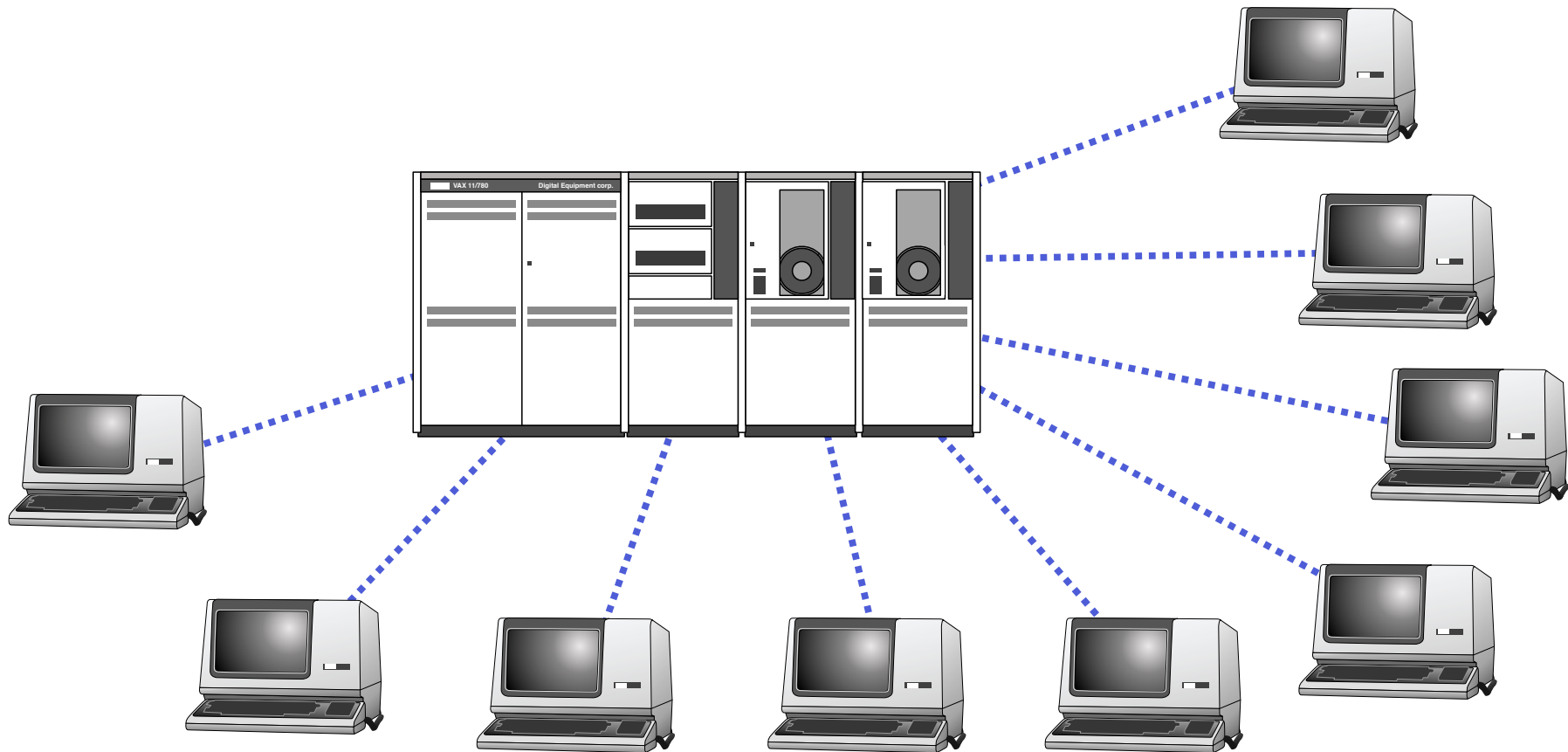
Round Robin による順繰り実行

- 複数アプリの同時実行

利用者の反応速度の方がコンピュータの動作時間より充分遅いので、見かけ上「同時並行」に見える

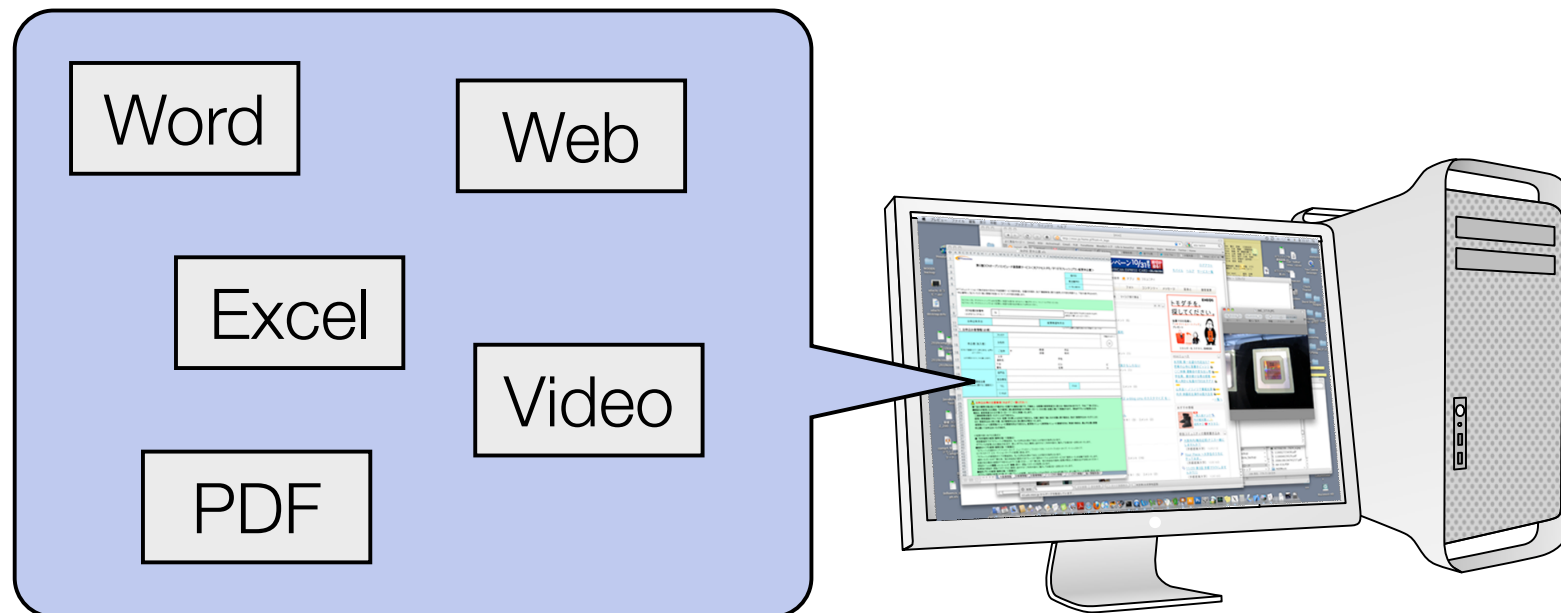
TSS (Time Sharing System)

- 大型ホスト + タム端末の組み合わせ
- 多数の利用者を専用端末が同時並行に操作



TSS (Time Sharing System)

- パソコンにおける TSS
- 一人の利用者が複数のアプリケーションを同時並行に実行
(動画を流しながら文書を作成するなど)



CPU 切り替え方式

- preemptive

タイマーによる切り替え

- non-preemptive

タイマーによらずプログラムが自主的に譲る

- 現在の Windows XP/Vista/7, MacOSX, Linux などはずべて TSS かつ preemptive (個人向けなので「端末」はない)

マルチプログラミングによる問題

- スケジューリング

優先順位づけ

- 排他制御

メモリやデバイス、ファイルに対するロック

デッドロック

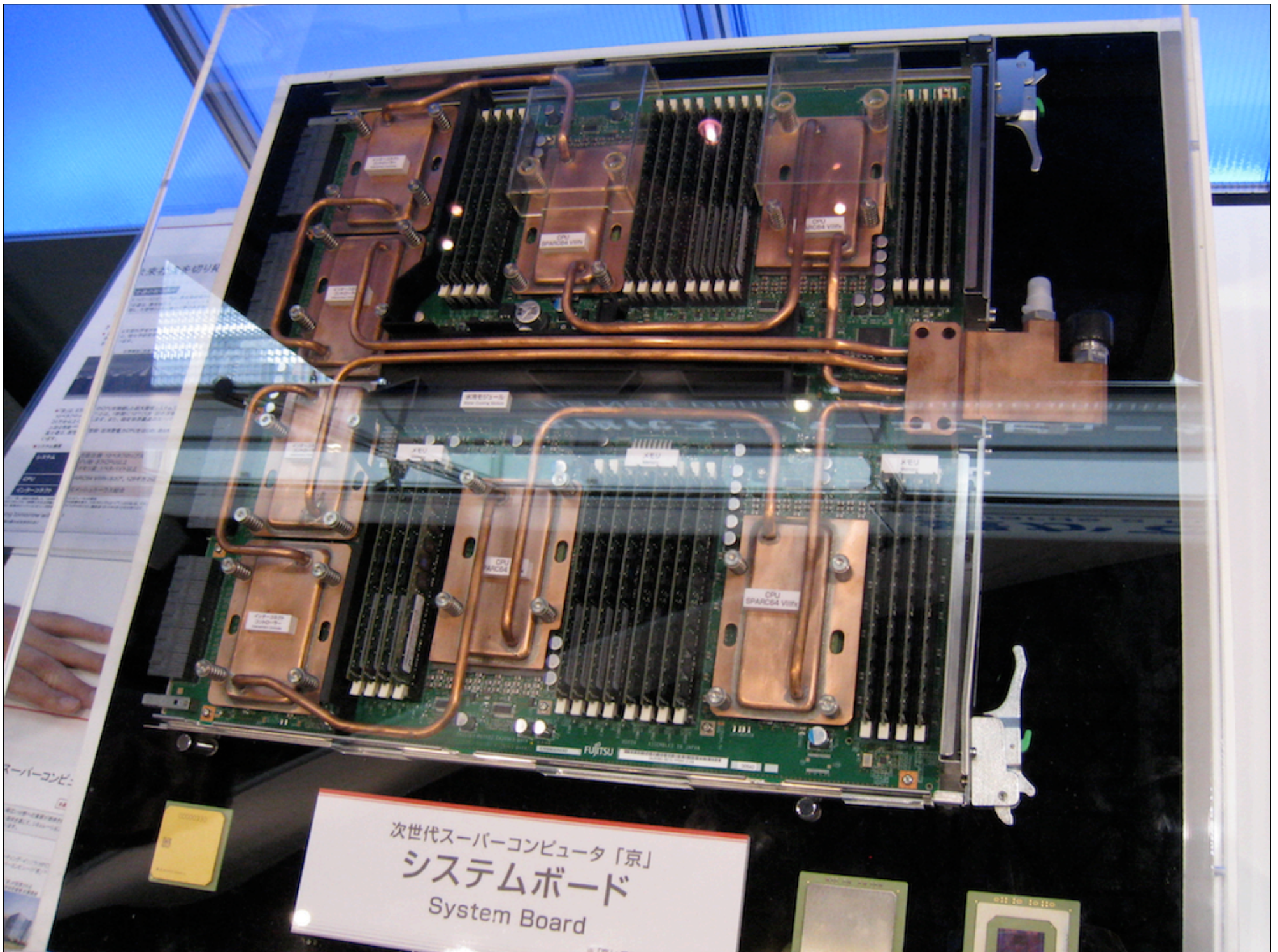
- メモリ管理＝再利用

将来

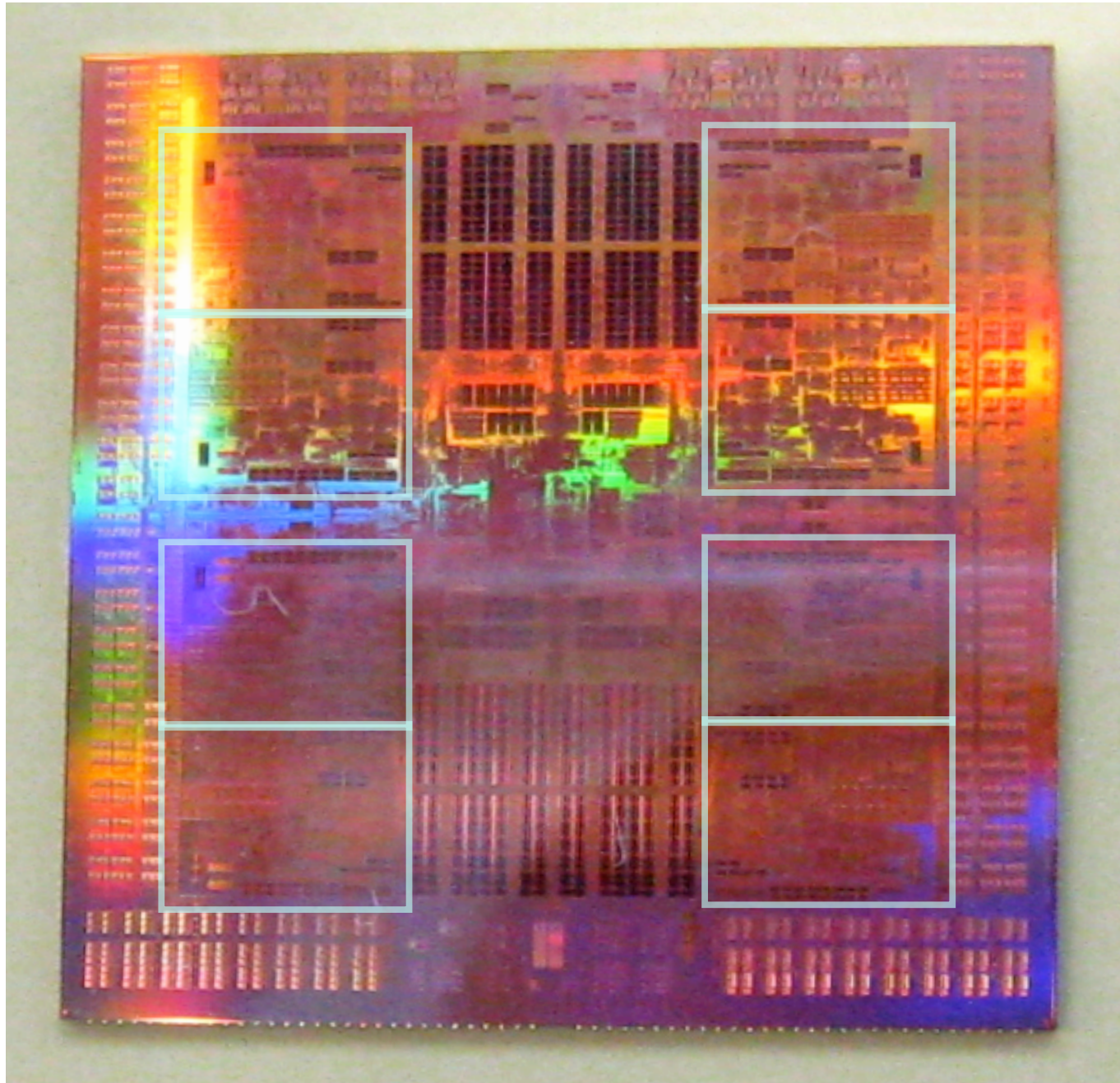
- マルチコア
- クラスタ



「京」スパコン, Fujitsu, at CEATEC 2010



次世代スーパーコンピュータ「京」
システムボード
System Board



SPARC64 VIIIfx, Fujitsu, at CEATEC 2010