

システムプログラム概論

Introduction to Systems Programming

プロセスとスケジューリング

門林 雄基

Youki Kadobayashi

NAIST

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

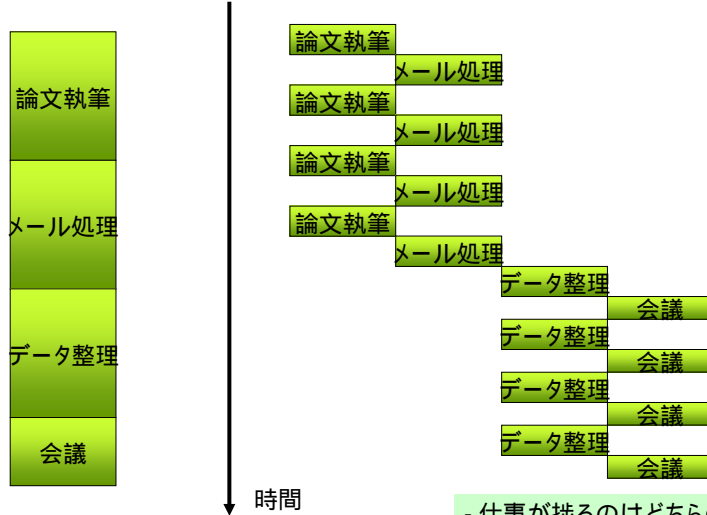
アウトライン

- スケジューリング
- マルチプログラミング
- プロセス

- プロセスの管理
- プロセスの生成と抹消
- プロセスの状態
- プロセスの優先度
- プロセスの切り替え
- プロセスの制御

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

複数の仕事を処理する二つの方法



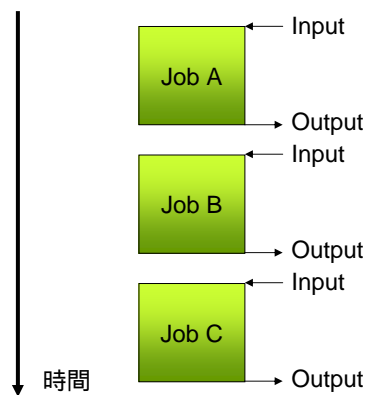
- 仕事が捗るのはどちらの方法か？
- 人を待たせないのはどちらか？

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

スケジューリング(1):ジョブスケジューラ

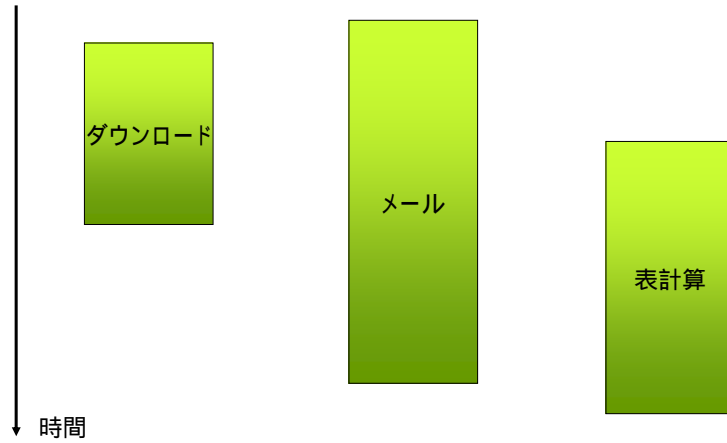
■ ジョブスケジューラ

- 今日ほとんど使われないが、スーパーコンピュータ等で残っている
- バッチ処理
- NQS, LSF等



Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

マルチプログラミング



- 複数プログラムの並行処理を単一のプロセッサで擬似的に実現する

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

スケジューリング(2): プロセススケジューラ

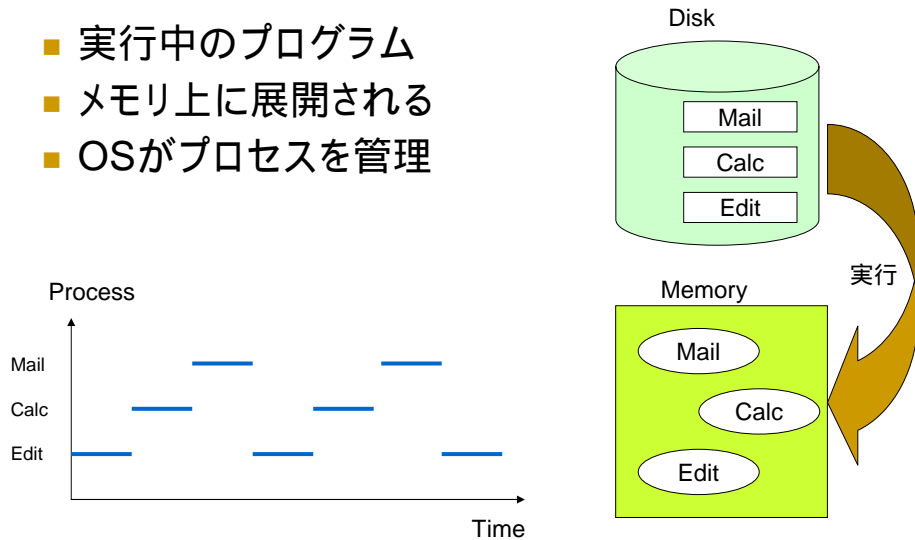
- プロセススケジューラ
 - マルチプログラミングを実現する基本的しくみ
 - 対話型処理では必須
 - プロセスとは？



Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

プロセス

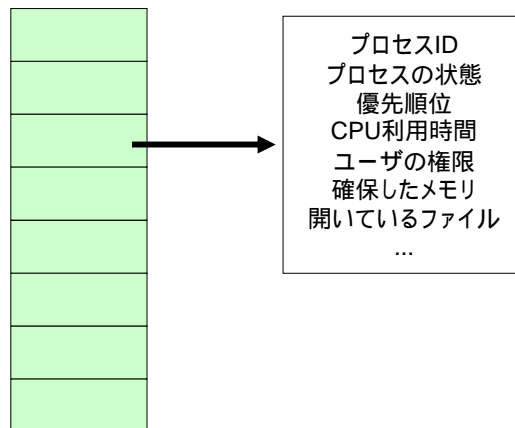
- 実行中のプログラム
- メモリ上に展開される
- OSがプロセスを管理



Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

プロセスの管理

- OSはプロセス一覧を保持
 - プロセステーブル



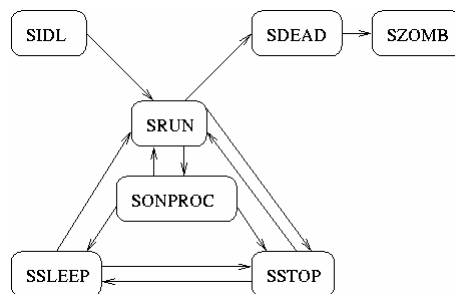
Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

プロセスの生成と抹消

- プロセスの生成：
 - メモリ領域を確保
 - ディスクからプログラムを読み込む(メモリ上に展開)
 - プロセステーブルへ新規プロセスを追加
 - ...
- プロセスの抹消：
 - 当該プロセスが使っていたメモリ領域を解放
 - プロセステーブルから当該プロセスを削除
 - ...

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

プロセスの状態



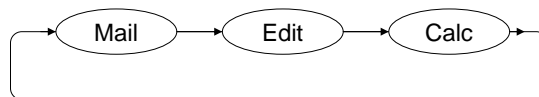
- SIDL: process being created
- SRUN: runnable
- SONPROC: being executed
- SSLEEP: awaiting an event
- SSTOP: process stopped
- SZOMB: process being terminated

NetBSD process states, from USENIX '02 paper by N. Williams

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

プロセスへのCPUの割り当て(1)

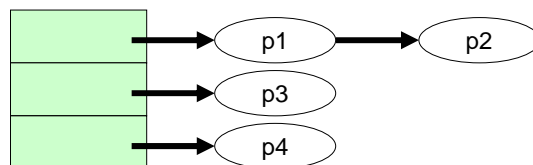
- ラウンドロビン・スケジューリング (round robin)
 - 実行可能プロセスのリストを維持
 - 順にCPU時間を割り当てる
- time slice
 - 各プロセスに割り当てられるCPU時間の単位



Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

プロセスへのCPUの割り当て(2)

- プライオリティ・スケジューリング
 - 例) 優先度ごとにリストを保持
 - 同じ優先度のプロセス同士ではラウンドロビン

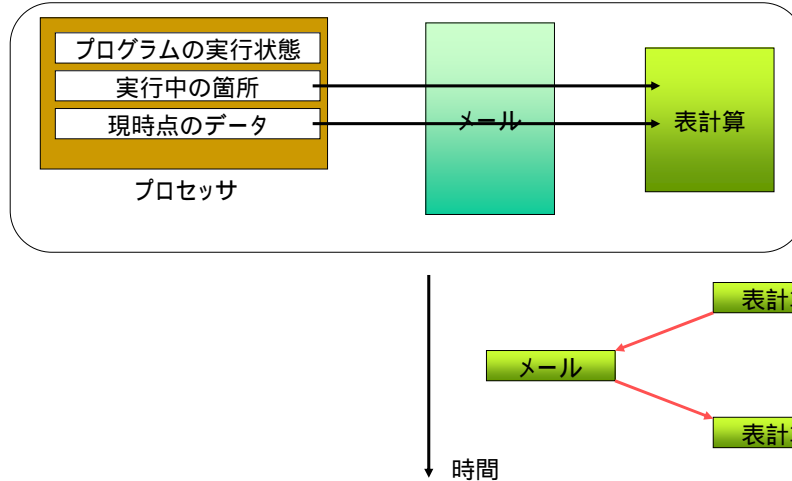


- 最も単純なもの 低い優先度のプロセスがまったく実行されない
 - 飢餓状態 (starvation) に陥る
 - 解決策 -- 例えば weighted round robin

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

プロセスの切り替え

■ コンテキストスイッチ



Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

プロセスの制御

■ 強制終了

UNIX:

SIGINT
SIGTERM

■ 一時休止、再開

SIGSTOP
SIGCONT

■ 状態監視

SIGUSR1
SIGINFO

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

考察1

- なぜジョブ・スケジューリングはほぼなくなったか？

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

考察2

- スケジューリング・アルゴリズムに求められる要件
 - 公平さ: 各プロセスに CPU 時間を公平に分け与える
 - 効率: 常に CPU 稼働率を 100%に維持する
 - 応答速度: 対話型処理の応答速度を速める
 - 回転率: バッチ処理の出力時間を速める
 - スループット: 一定時間内に処理できるジョブ数を最大化
- 時代とともに変わる

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.

まとめ

- スケジューリング
- マルチプログラミング
- プロセス

- プロセスの管理
- プロセスの生成と抹消
- プロセスの状態
- プロセスの優先度
- プロセスの切り替え: コンテキストスイッチ
- プロセスの制御: シグナル

Copyright(C) 2007 Youki Kadobayashi, NAIST. All rights reserved.