

リアルタイム処理

ハードリアルタイム処理

- 所定の期限内に処理を完了しなければならない
- 事前に処理時間を予測してスケジューリング
リアルタイムOSの例: VxWorks, pSOS, QNX, ITRON他

ソフトリアルタイム処理

- 所定の期限内に処理をするように努力する
- 時間制約に失敗した場合は例外処理を行う

QoS (Quality of Service) 処理

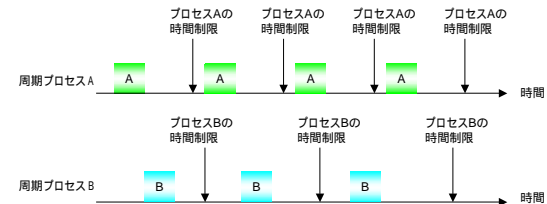
- サービス(計算、通信、音声、画質)の品質を保証する

Ichiro Satoh

リアルタイムスケジューリング

周期タスク(プロセス)のリアルタイムスケジューリング

一定間隔の制限時間内に処理を完了



リアルタイム処理に必要なシステム(センサーの監視、機械の制御など)では所定のタスクを周期的に実行することが多い

Ichiro Satoh

スレッド

プロセス中の処理の流れ

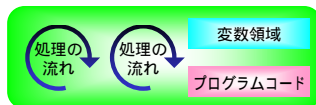
- 一つのプロセスに複数のスレッドが存在することができる
- スレッドが一つしかないとき従来のプロセスと一致する
- プログラム中の変数などは共有する

軽量プロセス (LWP: Light Weight Process) と呼ぶことができる

c.f. 従来のプロセスの切り替え(コンテキストスイッチ)ではタイムスライの6%の時間を消費 重量プロセス



シングルスレッドプロセス
(従来のプロセス)



マルチスレッドプロセス

Ichiro Satoh

スレッドのプログラム例

プログラム中の一つ以上の部分が同時に処理される

疑似プログラム:

```
program {  
  thread A {  
    for (int i = 0 ; i < 100 ; i++ ) {  
      printf("Thread A: %d\n", i);  
    }  
  }  
  thread B {  
    for (int j = 0 ; j < 100 ; j++ ) {  
      printf("Thread B: %d\n", j);  
    }  
  }  
}
```

同時実行

実行例1
(スケジューリングにより相違)
Thread A: 0
Thread B: 0
Thread A: 1
Thread B: 1
.....

Ichiro Satoh

スレッドの効用

例: 表計算ソフトウェア
ユーザインタフェースの利便性向上

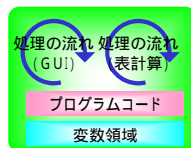
従来プロセス:

- 計算処理中はユーザとの対話処理は不可能
- 逆に対話処理中は表計算処理が不可能



マルチスレッドプロセス:

- 計算処理と対話処理を別のスレッド実行 (同時に実行可能)
- 同時に複数の処理が可能



Ichiro Satoh

スレッドの種類

ユーザレベルスレッド

- ユーザレベルのライブラリにより実現されるスレッド
- スケジューリング方法はユーザにより定義可能
- OSからは複数ユーザスレッドは一つのプロセスとして制御される

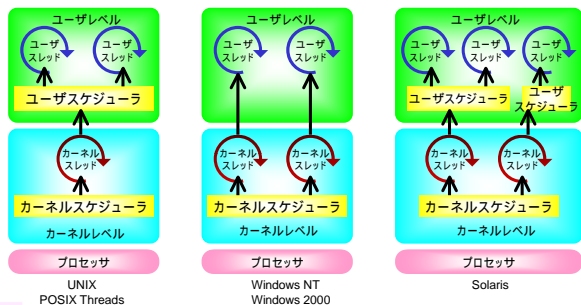
カーネルレベルスレッド

- OSにより高速にスケジューリングされるスレッド
- スケジューリング方法はOSにより定義 (ユーザによる変更は不可能)
- 各スレッドごとにOSに制御される

Ichiro Satoh

スレッドのスケジューリング

OSによりスレッドの実現方法は相違



Ichiro Satoh

スレッド間共有変数

同じプロセス内のスレッドは変数を共有可能

```
program {
    int x = 0;
    thread A {
        for (int i = 0 ; i < 100 ; i++ ) {
            int y = x;
            y = y + 1;
            x = y;
            printf("Thread A: %d\n", x);
        }
    }
    thread B {
        for (int j = 0 ; j < 100 ; j++ ) {
            int z = x;
            z = z + 1;
            x = z;
            printf("Thread B: %d\n", x);
        }
    }
}
```

実行順序例1
(スケジューリングにより相違)

```
int y = x;
y = y + 1;
x = y;
int z = x;
z = z + 1;
x = z;
```

実行順序例2
(スケジューリングにより相違)

```
int y = x;
int z = x;
y = y + 1;
x = y;
z = z + 1;
x = z;
```

コンテキストスイッチのタイミングにより
処理結果が異なることがある

Ichiro Satoh