

## 基礎OS③ オペレーティングシステムの構造

2012年度(3時限目)

### 問1 割込みとOS

入出力動作中に他のプログラムを実行することにより、CPUの使用効率を高める技術は、以下のどれか。尚、これを実現する重要な機構が割込みである。

- A. システムプログラミング
- ☒ B. 多重プログラミング
- C. Webプログラミング
- D. 構造化プログラミング
- E. アプリケーションプログラミング
- F. ネットワークプログラミング

多重プログラミングはCPUの使用効率を高めることを目的としている。入出力が始まると他のプログラムにCPUを切替え、実行中状態にする。また、入出力が終わったプログラムは、実行中にすることができる。これを実現するためには、入出力の開始、終了を検出する必要があり、その機構が割込みである。

### 問2 割込みの種類

外部割込みに分類されるのはどれか。(基本情報 平成17年度秋期 問20)

- A. 主記憶に存在しないページをアクセスしようとしたときに発生する割込み
- B. 入出力要求など、OSに対してサービスを依頼したときに発生する割込み
- ☒ C. ハードウェアが異常を検知したときに発生する割込み
- D. 浮動小数点演算でオーバフローが起こったときに発生する割込み

次の次のスライド参照

### 問3 割込みの種類

内部割込みの原因となるものはどれか。

- A. コンピュータの電源装置の故障
- ☒ B. 許可されていないメモリ領域へのアクセス
- C. 計算結果をメモリに格納する際の異常
- D. 設定された時間の超過
- E. ディスクへのアクセスが終了

次のスライド参照

### 重要: 割り込みとOSの起動

- ハードウェアがイベント(割込み原因)を検出。割込みによりOS(カーネル)を起動
  - そのとき実行していたプログラムを中断し、OS(カーネル)の特定番地にジャンプ
  - OS(カーネル)は、割込みによってのみ起動される

割込みの種類(注1)と割込み原因

- 外部割込み(external interrupt): ハードウェア割込みとも呼ばれる
    - ハードウェアが原因で発生する割込み
      - マシンチェック割込み: ハードウェアの故障、メモリの読み出しエラーなど
      - 入出力割込み: 入出力装置からの動作終了通知、異常通知など
      - 時計(タイマ)割込み: 設定された時間の超過
  - 内部割込み(internal interrupt): ソフトウェア割込みとも呼ばれる
    - プログラムの実行が原因で発生する割込み
      - プログラムの誤り: 演算例外(注2)、不正命令(注3)、メモリ保護違反(注4)
      - システムコール: アプリケーションプログラムによるシステムコールの発行
- 注1: 外部割込みを「割込み」、内部割込みを「割り出し(trap)」と言う人もいる  
- 注2: ゼロ除算、オーバフロー、注3: 定義されていない命令コードの使用  
- 注4: 許可されていないメモリ領域へのアクセス

### 問4 割り込み処理

割り込み処理において、ハードウェアがメモリの指定番地に設定し、割り込みの原因を伝えるものは以下のどれか。

- A. 命令コード
- B. 割込み命令
- ☒ C. 割込みベクトル
- D. エラー番号
- E. プログラムカウンタ
- F. ベースレジスタ

次の次のスライド参照

## 問5 割り込み処理

割り込み処理において、カーネルにジャンプしてきた後、割り込まれたプログラムを再開するために必要な情報を保存する処理は、以下のどれか。

- A. メモリ管理
- ☒ B. レジスタ退避
- C. スケジューラ起動
- D. ファイル管理
- E. 状態制御
- F. 同期制御

次のスライド参照

割り込みベクトル、割り込み分析、レジスタ退避、プログラムカウンタの意味を理解すること

## 重要: 割り込み処理

- ハードウェアの処理
  - ①プログラムの実行中に割り込みの事象が発生(これを検出)
  - ②**割り込みベクトル**(割り込み原因番号)をメモリの指定番地に設定
  - ③OS(カーネル)の指定番地(割り込み処理ルーチンの入り口)にジャンプ
- OS(カーネル)の処理
  - ④**レジスタ退避**(レジスタの内容<sup>(注1)</sup>をプログラム実行の管理エリア<sup>(注2)</sup>に保存)
  - ⑤**割り込み分析**により、原因ごとの割り込み処理を選択<sup>(注3)</sup>
  - ⑥選択した割り込み処理にジャンプし、実行を開始
- 注1: **プログラムカウンタ**(割り込まれた命令の次に実行するはずだった命令の番地)および各種汎用レジスタの内容。割り込まれたプログラムを再開する際、元の値に戻す必要がある。
- 注2: 実行中のプログラムは、**プロセス**として管理されている。この管理エリアは各プロセスに割り当てられ、**PCB**(Process Control Block)という。
- 注3: 割り込みベクトルの値をインデックスとした配列に割り込み処理プログラムの番地を記述したEDT(割り込みベクトルテーブル)からジャンプ先番地を取り出す。

## 問6 入出力制御

入出力装置がメモリを直接参照することで、データの転送処理からCPUを解放し、高い効率を実現する方式を何と言うか。

- ☒ A. DMA
- B. IDL
- C. IPC
- D. CDE
- E. TSS

DMAの処理

CPUは、入出力装置にデータの場所と指示を送る  
入出力装置は、**入出力制御装置**を介して、**メモリを直接参照**  
(CPUはデータの転送から解放され、プログラムの実行に専念できる)  
入出力の完了は、割り込みによって通知される

## 問7 入出力の効率化

通常は、レコードの長さは、ブロックより短く、そのままで入出力を行うと効率が悪い。レコードを複数まとめて1つのブロックとし、入出力回数を減らすことで効率を上げる方法は以下のどれか。

- A. キャッシング
- ☒ B. ブロッキング
- C. バッファリング
- D. マルチプレクシング
- E. スプーリング

アプリケーションプログラムは**レコード**単位に処理を行うので、**レコード**単位に入出力要求をしていく。  
アプリケーションプログラムが要求する度に、1レコードずつ入出力するよりも複数のレコードをまとめて**ブロック**とし、1回で入出力を行った方が効率が良い。  
(入出力の処理回数を減らせる)  
この方法を**ブロッキング**と呼ぶ。

ブロックはOSが入出力装置にアクセスする単位で  
ブロック長>レコード長

## 問8 ブロッキングの効果1

1レコード200バイトのデータを1200件出力する。1ブロック4096バイトでブロッキングする場合、ブロッキングしない場合と比べ、ディスクのアクセス回数が何%削減されるか。【数値(整数)のみを半角で記入】

答 95 (95%)

ブロック化係数  
 $4096 \div 200 = 20.48 \rightarrow 20$   
ブロック化しない場合のアクセス回数 = 1200  
ブロック化した場合のアクセス回数  
 $1200 \div 20 = 60$

削減されたアクセス回数  
 $1200 - 60 = 1140$   
削減率  
 $1140 \div 1200 = 0.95 = 95\%$

または、ブロック化したアクセス回数は、元のアクセス回数の何%かを計算  
 $60 \div 1200 = 0.05 = 5\%$   
削減率  
 $100\% - 5\% = 95\%$

## 問9 ブロッキングの効果2

レコード1000バイトのデータを1200件出力する。1ブロック4096バイトでブロッキングする場合、ブロッキングしない場合と比べ、ディスクのアクセス回数が何%削減されるか。【数値(整数)のみを半角で記入】

答 75 (75%)

ブロック化係数  
 $4096 \div 1000 = 4.096 \rightarrow 4$   
ブロック化しない場合のアクセス回数 = 1200  
ブロック化した場合のアクセス回数  
 $1200 \div 4 = 300$

削減されたアクセス回数  
 $1200 - 300 = 900$   
削減率  
 $900 \div 1200 = 0.75 = 75\%$

または、ブロック化したアクセス回数は、元のアクセス回数の何%かを計算  
 $300 \div 1200 = 0.25 = 25\%$   
削減率  
 $100\% - 25\% = 75\%$

## 問10 ブロッキングの効果3

1レコード200バイトのデータを25件出力する。1ブロック4096バイトでブロッキングする場合、ブロッキングしない場合と比べ、ディスクのアクセス回数が何%削減されるか。【数値(整数)のみを半角で記入】

答 92 (92%)

ブロック化係数

$4096 \div 200 = 20.48 \rightarrow 20$

ブロック化しない場合のアクセス回数=25

ブロック化した場合のアクセス回数

$25 \div 20 = 1.25 \rightarrow 2$

削減されたアクセス回数

$25 - 2 = 23$

削減率

$23 \div 25 = 0.92 = 92\%$

または、ブロック化したアクセス回数は、

元のアクセス回数の何%かを計算

$2 \div 25 = 0.08 = 8\%$

削減率

$100\% - 8\% = 92\%$