

## 第12回 割り込み・入出力アーキテクチャ

### [1] 割り込みとその処理

#### a. 教科書 9.1.1 を参考にして、空欄を埋めよ

割り込みは、実行中の処理を( )して、( )を行った後に、( )する機能である。

割り込みが発生すると、あらかじめ決められた( )へ分岐する。そして、割り込みルーチンの実行が終了したら、( )。したがって、通常ルーチンは中断するものの、結果としては( )見える。

#### b. 割り込みの説明として、正しいものを選び

- ① 割り込みは、実行中の処理が停止したとき、次の処理をCPUに割り付ける機能である
- ② 割り込みは、実行中の処理を一旦停止し、他の処理(プログラム)を行ったのち再開する機能である
- ③ 割り込みは、実行中の処理を一旦停止し、命令を1つだけ行ったのち元へ戻る機能である
- ④ 割り込みは、実行中の処理を中止し、他の処理を起動する機能である

#### c. 割り込みの原因による分類の説明で、間違っているものを選び

- ① 外部割り込みはCPUの外部で起こった原因、内部割り込みはCPUの内部で起こった原因である
- ② 内部割り込みは、命令の実行に伴って発生するものであるのに対し、外部割り込みは、(元の原因はプログラムにあるにしても)命令の実行やタイミングとは無関係に発生するものである。
- ③ 入出力装置が外部割り込みに含まれるのは、入出力装置の動作の進行がCPUのタイミングでは制御されておらず、CPUの動作とは無関係なタイミングで動作完了信号が発生するためである
- ④ 演算オーバーフローが内部割り込みに含まれるのは、CPUが演算命令を実行したためにオーバーフローが発生しそれをプログラムに通知する目的で割り込みが発生するからである

#### [補足] ことば

割り込み(Interrupt)を表 9.1 のように分類するとき、内部割り込みをわざわざトラップ(Trap)と例外(Exception)に分類している。これは、トラップはユーザ(プログラマ)が意図的に起こすものであるのに対し、例外は(プログラムによって起こるのだが)間違っただけ起きたものという考え方による。また、トラップ・例外に対して、外部割り込みのみをインタラプトと呼ぶ場合もある。余談だが、Java 言語で Exception という概念がある(構文 try, catch, throw など参照)が、割り込みの中の例外(Exception)と似たような概念である(同じ原因も含まれるが、Java の場合はもう少し上位のソフト寄りの原因も含む)。

#### d. 割り込み発生時の振舞いについて、間違っているものを選び

- ① 割り込みが発生すると、直前まで実行していた処理を中断し、あらかじめ指定したアドレスにジャンプする。
- ② 割り込み発生時にジャンプする先には、割り込みを処理するためのプログラム(割り込み処理プログラム、割り込みルーチン、割り込みハンドラー)が置かれる。割り込み種別によって飛び先を複数設けてある場合は、飛び先の一覧を「割り込みベクター」と呼ぶ。
- ③ 割り込み処理プログラムは、それまで実行していたCPUとは別の、割り込み処理専用のCPUで実行される。
- ④ 割り込み発生時に割り込み処理にジャンプする際、(割り込み処理の終了時に)元の処理に戻るために、割り込み発生時点でのプログラムカウンタ(=どこを実行していたか)をセーブ(格納)する。

#### e. 割り込みのマスクについて、間違っているものを選び

- ① 割り込みのマスクとは、割り込みの原因が発生しても、プログラムへの割り込み(=特定のアドレスへジャンプする動作)が起こらないように設定する機能である
- ② 割り込みの種類によって、マスクできる割り込みと、マスクできない割り込みがある。マスクできない割り込みは、たとえばハードウェアエラーなど緊急度の高い割り込みに用いる。
- ③ 割り込みに優先レベルが設定されているCPUでは、割り込みレベルが高いほど優先して処理される。割り込みが二重に発生する場合、たとえば、(ハードウェアエラー>入出力割り込み)であるとき、入出力割り込みの処理ルーチンの中で更にハー

ドウェアエラー割込みが発生すると、ハードウェアエラー割込みが受け付けられ、ハードウェアエラー割込みの処理ルーチンへ飛ぶ。

④ 同じレベルの割込みの場合、たとえば1つの入出力装置Aからの割込みを処理している最中に、別の入出力装置Bから割込みが発生した場合、優先レベルの例と同様に、Aのために実行している割込み処理ルーチンの中から更に(Bのために)割込み処理ルーチンへジャンプする。

[補足] 同じレベルの場合、前の割込み処理が完了するまで、(割込み処理プログラムの中でそのレベルの割込みマスクを設定する、もしくはハードウェア的に自動的にそのレベルの割込みマスクが設定される、などの方法で)、次の割込みが入れないようにする。割込み処理ルーチンを抜け出る直前に、割込みマスク状態もリセットして、次の割込みが入れるようにする。

f. 割込みの受付のタイミングについて、正しいものを選び

- ① 割込みの原因は1つの命令の実行の最中にも発生するので、発生した時点で実行中の命令を中断させて、割込み処理にジャンプする。
- ② 割込みの原因は1つの命令の実行の最中にも発生するが、一般には、発生時点で実行していた命令(1つ)は最後まで終了させ、次の命令に移る前に割込み処理にジャンプする。
- ③ 割り込みの原因は必ず1つの命令が終了したことによって起こるのであるから、その時点で割込むと、次の命令に移る前に割込み処理にジャンプする。
- ④ 割込みを受付けるタイミングは、いくつかのあらかじめ決まった命令の直後に限定されている。これらの命令は、次の命令を実行する前に割込み処理にジャンプしても、処理結果がおかしくならないような命令である。割込み処理はこれらの命令が出てくるまで待たされ、その直後に割込み処理にジャンプする。

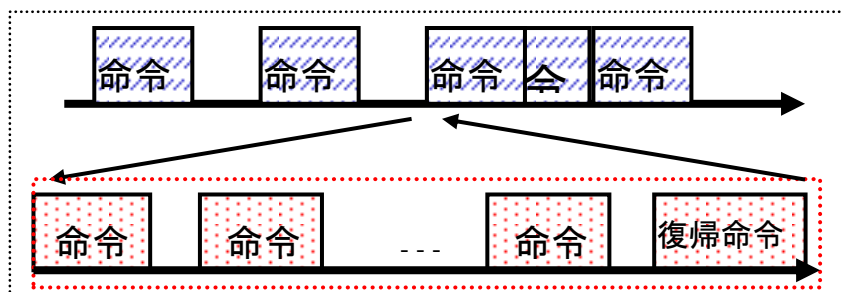
[補足] 割り込みの戻り先。

通常は、割込み処理の後の戻り先は、割込まれた命令の次の命令であるが、ページフォルト割込みだけは例外的に、割込まれた命令自身に戻る。これは、ページフォルトが起きた場合に実行中の命令が最後まで到達できないため(メモリがアクセスできない)で、ページフォルトを起こした命令自身を、命令の頭からやり直す。

[補足] 割り込み原因の特定

割込みは、さまざまな原因で発生するが、割込みベクターのエントリーの数は少ない(ハードウェアで決められている)ので、細かい原因の特定・区別は割込み処理プログラムの中で行う。言い換えれば、割込みベクターの種類の違いで区別できるのは大括りな分類に限られる。細かい原因は、特定のハードウェアレジスタの中のフラグビットとして読み出せるCPUが多い。また、入出力割込みでは、複数ある入出力機器の割込み信号線をすべてORしてCPUに接続するので、どの機器が割込みを起こしたかは機器に問合せないとわからないし、また機器の内部で何の原因で割込みを起こしたのか、動作完了なのか機器のエラーなのか、などは、機器内部のステータスレジスタを読み出さないとわからない。これらはいずれも割込み処理プログラム内で(ソフトウェアで)行う。

g. 右に示した、割込み時動作を説明するための図を用いて、割込み時の動作を説明してみよ。



h. 割込みを起こす原因を5つ挙げて、それぞれについて、名称、割込み原因の詳細な説明、内部割込みか外部割込みか、を示せ。

名称	割込み原因の詳細な説明	内部/外部の別


- i. 割込みのマスクという概念があるが、これはどういう考えか、説明せよ
- j. 割込みのレベルという概念があるが、これはどういう考えか、説明せよ
- k. 同一レベルの割込みが重なって起こることが有り得る。たとえば入出力割込みの処理中に入出力割込みがおきることが考えられるし、演算オーバフロー割込みの処理中に別のオーバフローが起こることも考えられる。その場合、ハードウェアはどのように対処するのか？ 説明せよ。
- l. 割込み処理ルーチンの行うべき処理はどのようなものか。①入力機器に於いて入力データがそろったことを示す「入力完了割込み」の場合には何をすべきか。②計算オーバフローが発生した場合の割込みの場合は、何をすべきか。③電源が失われたことを示す割込みの場合は、何をすべきか。

[2] 入出力の方式 (教 p. 107~)

a. 次の分の空欄を埋めよ

入出力制御方式(教 p.107~)には、直接制御方式と間接制御方式がある。直接制御方式は( ① )が( ② )装置を直接に制御する方式である。つまり、入出力装置と、メモリ又はレジスタの間のデータのやりとりを、( ③ )が命令によって行う。具体的には、( ④ )マップ方式の場合は( ⑤ )がメモリアクセス命令(ロード命令やストア命令、教科書ではMOV 命令)によって入出力機器との間でデータを移動する。それに対して( ⑥ )マップ方式の場合は( ⑦ )が入出力専用命令(IN 命令、OUT 命令など)を使って入出力機器との間でデータを移動する。いずれにしても1つ1つのデータのやり取りに必ず( ⑧ )が関わる。

それに対して、間接制御方式は( ⑨ )がメモリと入出力機器の間のデータのやり取りを行う。毎回の転送に CPU を使わないため、CPU の負担が( ⑩ )なる。教科書 p.108~110 にはDMAと入出力チャネルの2つの方式が説明してあるが、DMAはメモリと入出力装置の間のデータ転送を( ⑪ )が行う。これはCPUが( )に対して転送領域の先頭アドレスと長さ、及び転送の方向(入力か出力か)の情報を与えておき、起動を命じる。コントローラは(CPUとは独立して)入出力動作を行い、終了したら( ⑫ )をCPUに対して発生して終了を知らせる。DMAコントローラは通常は1つの領域から1回の入出力転送を行い、それが完了すると次の動作をCPUがセットし起動する。 それに対して、( ⑬ )はより複雑な入出力動作を制御するための専用回路を用意して、( ⑭ )の負担を更に軽減する。その回路はあたかもCPUと似たような複雑なものであり、チャネル専用の命令列(チャネルプログラム)を与えてそれに従って動作させることができる。チャネルはDMAに比べて、ハードウェアは( ⑮ )で高価になるが、CPUの負荷はより( ⑯ )なる。全体が高価なメインフレームコンピュータで広く使われたが、安価なPCでは使われていない。

b. 直接制御方式(プログラム入出力)・間接制御方式(DMA)について、誤っているものを選び

- ① 直接制御方式は、CPU内にあるハードウェアが入出力装置を直接制御するもので、命令からは「動作せよ」というと入出力装置を動作させ、結果を得てその命令が終了する。たとえば入力では、1つの命令によって一連の入力データ(たとえば1行分)が入力され、メモリに格納された後、その命令が終了する。
- ② 直接制御方式は、CPUがプログラム(一連の命令)によって入出力装置を制御するもので、入出力装置の細かい動作を命令で制御する。たとえば入力では、1つの命令は装置の起動、設定、1バイト分の入力などの個々の動作を制御するので、一連の入力データを取込むには繰り返して命令を出さなければならない。
- ③ 間接制御方式は、CPUとは別のハードウェアが入出力装置を制御するもので、入出力に伴うデータの転送に必要なCPUの負担を軽減する。

④ 間接制御方式では、CPUは入出力装置(もしくはチャネル方式の場合はチャネル)に対して入出力を命ずると、後の入出力動作は入出力装置で行い、一連のデータ転送が終わると入出力装置(またはチャネル)がCPUに割り込みを介して終了を通知する。

c. 入出力装置の制御(CPUからの接続)について、誤っているものを選び

① それぞれの入出力装置には、装置の動作を命ずるビットを持つレジスタ(CPUから書込み)や、装置の状態を示すビットを持つレジスタ(CPUから読出し)、出力データをCPUから与えるレジスタ(CPUから書込み)、入力データをCPUへ読み出すレジスタ(CPUから読出し)、などのレジスタがある。

② 上で述べたようなレジスタをCPUから指定するとき、それらがあたかもメモリアドレス空間の一部であるかのように見せて、読み書きはメモリに対する転送(ロード命令・ストア命令など)で行う「I/O マップトI/O 方式」と、それらは別の空間の一部として定義し、読み書きは特別な命令(入出力命令、Intel Pentium 系であれば inb 命令と outb 命令)で行う「メモリマップトI/O 方式」がある。この2つの方式のうちどちらを使うかは、CPUによって決まっている。

③ DMA方式では、(それぞれの入出力機器にある)DMAコントローラと呼ばれる回路が主記憶(メインメモリ)とのデータ転送を制御する。具体的には、転送したいブロックの主記憶上の開始アドレスと転送バイト数を、CPUがDMAコントローラに書込み、DMAコントローラに転送を命ずると、指定した主記憶上の領域から読出す、もしくは領域へ書込む。

④ 入出力チャネルによる方式では、「チャネル」が第2のCPUとして動作する。CPUはあらかじめ主記憶上の領域にチャネルが実行するプログラム(チャネルプログラム)を準備し、その領域のアドレスをチャネルに通知する。チャネルはその領域からチャネルプログラムを読み出して、逐次実行する。チャネルプログラムは配下の入出力機器の動作を制御し、主記憶上のデータを出力したり、入力したデータを主記憶上に格納する。これらの動作のための主記憶アクセスはDMAとして行われる。

## 《解答》

[1]

- a. 停止(中断)      他の処理 (割り込み処理)      元の処理を再開(元の処理へ復帰)  
割り込みルーチン (割り込みアドレス)      元の通常ルーチンへ復帰する  
通常ルーチンと割り込みルーチンが並行して実行されたように (通常ルーチンから見るとあたかも割り込みが無かったかのように)
- b. ②      ③は「命令を1つだけ」という点が誤り、④は「中止し」という表現が誤り(中断であってやめてしまうのではない)
- c. ①      CPUの内部・外部という区別ではなく②で言うように「命令が原因」で区別する。教科書では「機械語命令に同期している」か否かの違いと言っている。「同期」の意味は、機械語命令と同じタイミング、ということで、③入出力装置の動作は機械語命令と同じタイミングというわけではない(たとえば人がキーを押す)。④演算のトラブルは、その演算の機械語命令を実行する時に発生する。
- d. ③
- e. ④      1つの入出力割り込みの処理中に別の入出力割り込みが起こった場合、後から起きた割り込みは待たせておく。但し異なる種類の割り込みでレベルが違う場合、例えば入出力割り込みの処理中に、CPUやメモリのハードウェア障害の割り込みが起こると、入出力割り込み処理のプログラムが正しく動作していないことになるから、中断すべきである。だから、入出力割り込みよりハードウェア障害の割り込みの方が優先度が高く設定されている。
- f. ②      9.2.2節の図9.5にあるように、②が正しい。但し1つ例外があって、仮想記憶で使う「ページフォルト」の割り込みは命令の実行中に受付・処理され、その命令の先頭へ戻って再開する。つまりこの場合だけは、その命令が始まってから割り込み発生までの命令内の処理はすべてなかったことにされ、割り込みから復旧する時に命令の先頭からやり直す。
- g. 青で示した上段の流れは通常の(割り込みの無い場合の)命令の実行である。この途中で第2命令終了後に割り込みが受け付けられた場合、赤で示した下段の「割り込み処理ルーチン」の先頭へジャンプする。割り込み処理ルーチンの一連の処理が終了したとき、最後に復帰命令を実行することによって、割り込み発生時に記憶しておいた「戻り番地」へ復帰するこ

とができる。このとき、青で示した上段の流れは、赤で示した割り込み処理ルーチンをあたかも実行しなかったかのように、第2命令から第3命令へ継続したように実行されることが、大事なポイントである。そのために、割り込み処理ルーチンで使用する汎用レジスタ等の値は割り込み処理ルーチンの先頭で退避され、最後に復帰する直前に復旧されなければならない。

h.

名称	割り込み原因の詳細な説明	内部/外部の別
演算オーバーフロー	計算結果がオーバーフローを起こした	内部
アドレスエラー	CPU が要求したアドレスが正しい範囲外であった	内部
未定義命令	命令コードが定義されたものでなかった(プログラムでない部分を命令フェッチした)	内部
ページフォルト	仮想記憶に於いてアクセスしたアドレスのページが主記憶にロードされていなかった	内部
入出力割り込み	入出力機器からの動作完了を示す信号	外部
ハードウェア障害	ハードウェアの障害(内部情報のパリティエラー、電源断など)	外部
リセット割り込み	リセットボタンが押されたことを示す信号	(外部)*

(\*)リセット割り込みは教科書では内部・外部の別とは分けてあるが、外部と見ることもできる。

- i. 割り込みのマスクは、割り込みが発生するのを禁じる、つまり割り込み信号を受け付けないようにすることである。たとえば同じレベルの割り込みは1つ起きるとその次のものは受け付けないようにマスクしてしまうのが普通である。また、緊急度の高い割り込みは緊急度の低い割り込みをマスクしてしまうのが普通である。更に、プログラムが命令によって割り込みを禁止することができ、オペレーティングシステム上でタスク切替が起きないようにするのに使われる。  
なお、緊急度の高い割り込みはマスクできない(ノンマスクابل割り込み)のが普通である。
- j. 割り込みは、緊急度によってレベル分けされ、緊急度レベルの高い割り込みは低い割り込みをマスクする。たとえば、入出力割り込みはハードウェア障害割り込みより緊急度が低く、ハードウェア障害割り込み発生時にはマスクされる。一般に緊急度最大の割り込みは電源断の割り込みである。
- k. 同レベルの割り込みが重複して発生した場合には、現在処理中の割り込みの処理が終了しない限り、後から発生した割り込みはマスクされる。従って、後から発生した割り込みは、前の割り込みから復帰した直後に割り込む。これによって、後から発生した同一レベルの割り込みは待たされ、順次処理されることになる。
- l. ①入力完了割り込みは、入力データが入力機器上にそろったことを示すので、その割り込み処理では機器から入力データを CPU 側へ読み込み、必要なデータ処理を施し、元々入力を要求したプログラム内のバッファ領域へデータを渡す。割り込みから復帰するとそのプログラムが動作を継続するので、そこで入力結果を利用するコードを実行する。  
②ユーザのプログラムの処理の中で計算オーバーフローが発生した場合、通常はそのプログラムを異常終了させ、処理はオペレーティングシステムへ復帰し、次の(他の)プログラムを実行させる。ただし、Java 言語などでは、オーバーフローによる異常をユーザプログラム内で catch し、ユーザがプログラム内に用意した異常を処理するルーチンを起動させることができる。  
③電源が失われた場合、電源装置内のコンデンサ等に残る電荷のおかげでごく短時間ではあるが処理が可能なシステムもあり、その場合は主記憶上の結果をハードディスク等にセーブして残す処理をする。また、蓄電池によるバックアップ電源システムが用意されている場合、電源断の割り込みによりシステムの正常終了処理を起動する。

[2]

- a. ① CPU ② 入出力 ③ CPU ④ メモリ ⑤ CPU ⑥ I/O ⑦ CPU

- ⑧ CPU ⑨ 入出力専用のハードウェア(DMAコントローラ) ⑩ 軽く ⑪ DMAコントローラ  
⑫ 割り込み ⑬ 入出力チャネル ⑭ CPU ⑮ 複雑 ⑯ 軽く
- b. ① 直接制御方式では、1つの命令では入力装置の1つの動作、たとえば1文字分の読み込みが行われる。1行分を読むことはできない。(転送の単位が通常は1バイトであるから、1命令で1文字分しか読めない)
- c. ② ②では、メモリマップト方式とI/Oマップト方式が逆になっている。

《基本情報処理技術者試験問題から類似・関連問題》

- 1) 内部割り込みに分類されるものはどれか。(基本 20 秋 18)
- ① 商用電源の瞬時停電などの電源異常による割り込み
  - ② ゼロで除算を実行したことによる割り込み
  - ③ 入出力が完了したことによる割り込み
  - ④ メモリパリティエラーが発生したことによる割り込み
- 2) 外部割り込みに分類されるものはどれか。(基本 19 秋 20)
- ① インターバルタイマによって、指定時間経過時に生じる割り込み
  - ② 演算結果のオーバフローやゼロによる除算で生じる割り込み
  - ③ 仮想記憶管理において、存在しないページへのアクセスによって生じる割り込み
  - ④ ソフトウェア割り込み命令の実行によって生じる割り込み
- 3) 割り込みに関する記述のうち、適切なものはどれか。(基本 17 春 19)
- ① CPU は割り込みを受け付けると実行中のプログラムを中断し、プログラムの再開に必要な情報を磁気ディスクの特定の領域に格納する。
  - ② アプリケーションは、常に割り込みの発生を感知する必要がある。
  - ③ 入出力装置からの動作完了の通知は、内部割り込みに分類される。
  - ④ 複数の割り込みの発生に備え、個々の割り込み原因には優先順位が付けられる。

《解答》

- 1) ② 電源異常、入出力完了、メモリハードエラーはいずれも外部割り込みとみなす。
- 2) ① インターバルタイマは入出力機器と同じように見なすとよいだろう。CPUチップ内にあるが、外部割り込みとみなす。
- 3) ④