

⑦保護とセキュリティ(1)

高度OS 2015年度

問1 CPU保護

CPU保護の説明として不適当なものはどれか。  
ヒント: プログラムのバグによる無限ループからCPUを保護する。

- A. 特定のプロセスにより, CPUが独占されることから保護する。
  - B. CPUバーストが長時間続けば, そのプロセスを終了させる。
  - C. プロセスを実行中状態にする時に, タイマを設定する。
  - ☒ D. プロセスの不正な命令により, CPUが暴走することから保護する。
- Aは, CPU保護の目的を説明(正しい)。  
B, CはCPU保護の方法を説明(正しい)。

問2 特権命令

以下の説明のうち, 誤っているものはどれか。  
ヒント: 非特権命令とは, 特権命令以外の命令。特権命令はOS(カーネル)だけが実行できる。

- A. 非特権モードでは, 非特権命令だけが実行できる。
- ☒ B. 非特権命令は, 非特権モードのときにだけ実行できる。
- C. 特権モードでは, 特権命令と非特権命令が実行できる。
- D. 特権命令は, 特権モードのときにだけ実行できる。

特権モード: 全ての命令が実行できる(特権命令+非特権命令)  
非特権モード: 非特権命令だけが実行できる(特権命令は不可)  
従って, Aが誤り(非特権命令は, 特権モードでも実行できる)

問3 割り込み(内部割り込み)

プログラム割り込み(内部割り込み)の原因となり得るものはどれか。(第2種平成12年度・秋期, 平成9年度・秋期 改)  
ヒント: 非特権モードで特権命令を実行するとどうなるかを考えよ。

- A. プログラムの実行中にハードウェアが故障した。
- B. プログラムの実行時間が設定時間を超過した。
- C. プログラムで要求されたI/O処理が終了した。
- ☒ D. プログラムの実行中に不正命令を実行した。

A, B, Cは, 何れも外部割り込み(ハードウェア割り込み)。  
Dは, プログラムのエラーなのでプログラム割り込み(内部割り込み)の原因となる。  
非特権モードでは, 特権命令は不正な命令であり, 実行してはならない。

問4 入出力保護

アプリケーションが特権命令を実行した場合に発生する割り込みの種別は以下のどれか。

- A. 外部割り込み(不正命令)
- B. 外部割り込み(システムコール)
- C. 外部割り込み(I/O完了)
- ☒ D. 内部割り込み(不正命令)
- E. 内部割り込み(システムコール)
- F. 内部割り込み(I/O完了)

非特権モードにおいて実行された特権命令は不正な命令である(OSは, このプロセスを終了させる必要がある)  
不正命令やシステムコールはプログラムの実行が直接の原因であるので, 内部割り込みである(プログラムの中で発生)。  
I/O完了は, ハードウェアの動作が原因なので外部割り込みである(プログラムの外で発生)

問5 入出力保護

アプリケーションがI/O要求のシステムコールを発行した場合のOSの処理として最適なものは以下のどれか。

- A. アプリケーションプロセスを終了状態とする。
- B. アプリケーションプロセスをレディ状態とする。
- C. アプリケーションプロセスを実行中状態とする。
- ☒ D. 割り込み分析の結果, 装置ドライバを起動する。

この後, 装置ドライバの中で特権命令であるI/O命令を実行する。  
入出力処理が終わるまで, I/O要求をしたプロセスを待機状態とし, 代わりのプロセスを実行中状態とする。

高度OS⑤(7)

問6 入出力保護

入出力保護の説明として、不適当なものはどれか。

A. OSの装置ドライバによりI/O要求の正当性を検査する。

B. I/O要求のシステムコールを発行すると割込みが発生する。

C. アプリケーションは非特権モードでI/O命令を実行する。

D. OSは、特権モードでI/O命令を実行する。

C. I/O命令は特権命令であるので、非特権モードで実行すると不正命令となる。  
その結果、アプリケーションは終了させられる。(誤り)

B. アプリケーションが入出力をしたいときは、システムコールにより、OSに依頼する。  
OSを起動するために割り込みが発生するが、システムコールが原因であるので、エラーではない。(正しい)

A. D. 入出力保護の方法を説明(正しい)

高度OS⑦(7)

スライド(問7の添付ファイル)

問7 試行結果 ① 0を設定したら、一切のアクセスができなくなりました。  
② 5を設定したら、読取りと更新はできたが、実行ができなかった。  
③ 7を設定したら、すべてのアクセスができるようになった。

問8～10 セマフォと排他制御のプログラム

セマフォの定義

P(S):

Sの値を1減らす;

S≥0→nop;

S<0→発行元プロセスを待機状態に;

戻る;

V(S):

Sの値を1増やす;

S>0→nop;

S≤0→待機プロセスをレディ状態に;

戻る;

生産者プロセス:

(以下を繰り返す)

/\*ディスクのデータを変数nextplに読み込む\*/

(1)read(disk, nextp);

(2)P(b2);

(3)buffer[in]=nextp;

(4)in=(in+1) % n;

(5)V(b1);

消費者プロセス:

(以下を繰り返す)

(6)P(b1);

(7)nextc = buffer[out];

(8)out=(out+1) % n;

(9)V(b2);

(10)write(printer, nextc);

/\*変数nextcのデータをプリンタに印刷\*/

高度OS⑤(7)

問7 ファイルのアクセス制御

ファイルに、読取り、更新、実行という3種類のアクセス権を設定できる OS がある。この3種類のアクセス権は、それぞれに1ビットを使って許可、不許可を設定する。この3ビットを8進数表現0～7の数字で設定するとき、スライド(添付ファイル)に示した〔試行結果〕から考えて、適切な記述はどれか。(基本情報 平成17年度・春期改)

〔試行結果〕

① 0を設定したら、一切のアクセスができなくなりました。

② 5を設定したら、読取りと更新はできたが、実行ができなかった。

③ 7を設定したら、すべてのアクセスができるようになった。

A. 2を設定すると、実行だけができる。

B. 3を設定すると、更新だけができる。

C. 4を設定すると、読取りと作成ができる。

D. 6を設定すると、読取りと更新ができる。

①:000 すべて不許可

③:111 すべて許可

→「1」が許可、0が不許可

②:101 実行のみ不許可

↑↑↑

更新、読取り

実行 どちらかは不明

010=2を設定すると実行のみ可

高度OS⑤(7)

問8 セマフォ

スライド(問7の添付ファイル)に示すようなセマフォを用いてバッファの排他制御を行う2つのプロセスがある。変数in、outはバッファのポインタであり初期値はin=0、out=0。また、nはバッファ(buffer[ ])の数を表し、n=3とする。生産者は、(1)のディスクread中、消費者は(10)のプリンタwrite中で、それぞれ待機状態。セマフォ変数の値はb1=0、b2=3である。このあと、(A)生産者、(B)消費者、(C)消費者、(D)生産者の順で実行中状態となるものとし、その途中経過を考える。

(A)の実行が終わり、次に(B)で消費者が(9)の処理を行った後のセマフォ変数の値と消費者の状態は何か。[b1、b2、消費者の状態に半角のコンマ「,」で記述し、変数値(半角数字)と状態名(全角文字)を縦向きに記入。例: 3,レディ]

答 0,3,実行中 (b1=0、b2=3、消費者は実行中)

セマフォ変数b1が入力済みバッファ数、b2が空きバッファ数を示す。  
V命令を発行したプロセスの状態は変わらない。OSは(9)のV命令の処理が終わると発行元の消費者に戻る。すなわち、消費者は実行中。  
この後、(10) write命令により、OSは消費者を待機状態にする。  
(状態遷移は次々スライド参照)

高度OS⑤(7)

問8初めの状態までの処理例

以下のような実行によって、問8の初めの状態になる。

実行サイクル	事象の発生と命令の実行	生産者	消費者	b1	b2
	生産者、消費者生成	レディ	レディ	0	3
(a)生産者	生産者にCPU割り当て	実行中	レディ	0	3
	(1) read(disk, nextp)=I/O要求	待機	レディ	0	3
(b)消費者	待機 CPU割当て	待機	実行中	0	3
	(6)P(b1)=事象待合せ	待機	待機	-1	3
(c)生産者	read完了。CPU割当て	実行中	待機	-1	3
	(2)P(b2)、(3)、(4)	実行中	待機	-1	2
	(5)V(b1)=消費者事象発生	実行中	レディ	0	2
	(1)read(disk, nextp)=I/O要求	待機	レディ	0	2
(d)消費者	CPU割当て、	待機	実行中	0	2
	(7)、(8)、(9)V(empty)	待機	実行中	0	3
	(10)write(printer, nextc)=I/O要求	待機	待機	0	3

高度OS⑤(7)

問8、9の解説

入力済 空き

実行サイクル	事象の発生と命令の実行	生産者	消費者	b1	b2
	[定常状態]	待機	待機	0	3
(A)生産者	read完了。CPU割り当て	実行中	待機	0	3
	(2)P(b2)、その後(3)(4)	実行中	待機	0	2
	(5)V(b1)	実行中	待機	1	2
	(1) read(disk, nextp)=I/O要求	待機	待機	1	2
(B)消費者	write完了。CPU割り当て	待機	実行中	1	2
	(6)P(b1)、その後(7)(8)	待機	実行中	0	2
	(9)V(b2)	待機	実行中	0	3
	(10) write(printer, nextc)=I/O要求、	待機	待機	0	3
(C)消費者	write完了。CPU割り当て	待機	実行中	0	3
	(6)P(b1)	待機	待機	-1	3

問8 V命令を発行した時点では、発行元は実行中のまま。  
これに続いて、(10)を実行すると、OSはreadが済むまで消費者を待機状態にする。

問9 実行サイクル(B)で入力済バッファが無くなっている(b1=0)。さらに、(C)の(6)によりb1=-1となり、消費者が待機状態となる。

2

高度OS2019(7)

問9 セマフォ

前問に続いて、(C)で消費者が(6)のセマフォのシステムコールを発行し、その処理が行われた後のセマフォ変数の値と消費者の状態は何か。【b1, b2, 消費者の状態の欄に半角のコンマ「,」で区切り、変数値(半角数字)と状態名(全角文字)を順に欄に記入。(例 3,4レディ)】

答 -1,3,待機 (b1=-1, b2=3, 消費者は待機状態)

状態遷移は前スライド参照。

実行サイクル(B)によって、b1=0(入力済みバッファが無い)になっている。この後、消費者が実行しようとするば、止めなければならない。(C)の(6)P命令で、b1=-1となったので、OSは消費者を待機状態にする(P命令は消費者に戻らない)。OSの処理は、ここでいったん終了する。(他に実行可能なプロセスがある場合は、そのプロセスを実行中にする。)

この後、read完了の割り込みでOSが再度起動され、CPU割り当てを行って、生産者を実行中にする(実行サイクル(D))。問10のV命令でOSは消費者をレディ状態にする。生産者がread命令で待機状態になると、OSは消費者を実行中にする(実行サイクル(E)=(C)の残り)。ここで、P命令が消費者に戻り、(7)以降の命令が実行される。(次スライド参照)

高度OS2019(7)

問10 セマフォ

前問に続いて、(D)で生産者が(5)のシステムコールを発行し、OSの処理が終わった後のセマフォ変数の値と消費者の状態は何か。【b1, b2, 消費者の状態の欄に半角のコンマ「,」で区切り、変数値(半角数字)と状態名(全角文字)を順に欄に記入。(例 3,4レディ)】

答 0,2,レディ (b1=0, b2=2, 消費者はレディ状態)

生産者の(5)V命令により、b1=0となったため、OSは消費者をレディ状態にする。(OSの処理終了後、生産者に戻り、生産者は(1)を実行。)

実行サイクル	事象の発生と命令の実行	生産者	消費者	b1	b2
	問9の最後	待機	待機	-1	3
(D)生産者	read完了, CPU割り当て	実行中	待機	-1	3
	(2)P(b2), その後(3)(4)	実行中	待機	-1	2
	(5)V(b1)=[消費者]事象発生	実行中	レディ	0	2
	(1) read(disk, nextp)=I/O要求	待機	待機	0	2
(E)消費者	CPU割り当て, その後(7)(8)	待機	実行中	0	2
(C)の残り	(9)V(b2)	待機	実行中	0	3
	(10) write(printer, nextc)=I/O要求,	待機	待機	0	3

高度OS2019(7)

問8～10 各実行サイクル終了時の状態

実行サイクル	命令の実行	生産者	消費者	b1	b2
	[定常状態]	待機	待機	0	3
(A)生産者	(2)(3)(4)(5)(1)	待機	待機	1	2
(B)消費者	(6)(7)(8)(9)(10)	待機	待機	0	3
(C)消費者	(6)←ここで中断	待機	待機	-1	3
(D)生産者	(2)(3)(4)(5)(1)	待機	レディ	0	2
(E)消費者	再開後に(7)(8)(9)(10)	待機	待機	0	3

3