

記述らしき問題抜粋

—————アルゴ—————

(2013 1)

- ・安定なソートアルゴリズムとは何か？

—————アーキ—————

(2016 2013)

- ・二つの参照局所性の説明とキャッシュメモリに与える影響を答えよ
- ・キャッシュメモリとメインメモリで階層を形成することの利点をその理由とともに述べよ。

(2015 2)

- ・2進数の浮動小数点表現では、0.1 を誤差なく有限桁で表現できない理由を答えよ

(2014 2)

- ・計算機において符号付き整数の表現に2の補数系表現を用いることの利点を述べよ
- ・セマフォにおいてテストとセットを不可分操作とする必要性を述べよ

(2013 2)

- ・ブロックサイズが小さい時とブロックサイズが大きい時に平均メモリアクセスが大きな値となる理由をそれぞれ説明せよ

(2011 2)

- ・プロセッサを設計する際に遅延時間を決定する上での制約について答えよ
- ・ファイルサイズがハードディスク装置の使用効率とファイル平均転送時間に及ぼす影響について下記の観点から考察し簡潔に述べよ。  
①ファイルサイズとブロックサイズの関係 ②ブロックサイズとシーク回数

(2010)

プロセスの相互排除に必要な条件は何か。

(2009)

LRU と FIFO に対して、「実装の容易さ」「ブロック枠の数とヒット率の関係」の観点で比較したうえで、それぞれの特徴を理由とともに述べよ。

(2008)

最短シーク順（現在のヘッドの一に最も近いシリンダへのアクセスを優先させる）よりエレベータ順（ヘッドを移動させる方向を決め、その先にアクセス要求がなくなるまでその方向にヘッドを移動させながらアクセスを行い、その後、逆方向にヘッドを移動させアクセスを続ける）のほうがアクセス時間の観点で優れている点を 100 字で述べよ。

---

計算論

(2015 3)

・〜〜〜うにやうにや。状態集合  $Q$  上の二項関係が同値関係であることを証明せよ。

(2013 3)

・プログラミング言語の文法としてなんとか文法とかんとか文法ではなく文脈自由文法を用いるのは何か。

(2012 3)

チョムスキー標準形とは何か。

※正則言語、文脈自由言語とは何か。

(2011 3(9))

文脈自由言語が「曖昧」とはどういうことか。

# 1 アーキ穴埋め

## 1.1 OS 期末

候補語：プリページング、スワップイン、スワップアウト、デマンドページング、  
アドレス変換機構、ハードウェア機構、ソフトウェア機構、コンパイラ機能、  
記憶保護機構、動的、静的、開始時、実行時、終了時、キャッシュメモリ、  
メインメモリ、補助メモリ

ページング方式においては、どのタイミングでページを仮想アドレス空間から実アドレス空間に読み込むかが重要であり、大別して (a) と (b) がある。

(a) では、プログラム自身が実行時にアクセスや参照を要求するページを読み込む。すなわち、OS 機能を (c) が支援する (d) ページングである。長所としては、必要なページだけを読み込むため無駄がないことが挙げられる。一方、短所としては、プロセス実行の (e) にページフォールトが集中発生することが挙げられる。

一方、(b) は、アクセスや参照があると予測されるページを OS が実行前にまとめて読み込むものであり、OS 機能を (f) が支援する (g) ページングである。長所として、予測的中率が高ければ、複数ページを一度に読み込むブロック転送が適用できるので高速化できることが挙げられる。一方、短所としては、不要なページも読み込んでしまう可能性がある、予測的中率が低いとそのオーバーヘッドが顕在化する、などが挙げられる。

現代では、(h) の実装コストが低くなったので、余分で冗長なページを読み込んでおいても予測がはずれた場合の影響は少ないため、(b) を採用する OS が多い。

## 1.2 2016

(1) 計算機 (computer), 特に, 中央処理装置 (CPU: central processing unit) に関する以下の各小問に答えよ。解答は全て解答用紙の太線内に書くこと。

(1-1) 以下の文章の空欄 (a)～(e) に当てはまる最も適切な語句を, 下記の選択肢から選び, 記号で答えよ。

計算機の構成方式のことを (a) と呼ぶ。現在, 実用的に利用されている大部分の計算機は, 線形アドレス空間 (linear address space) を有するメインメモリ (main memory) 上にプログラム (program) 及びデータ (data) を置き, プログラムを逐次実行する, 等の (a) を採用している。このような計算機は, (b) 計算機と呼ばれる。

(b) 計算機では, (c) が示すメインメモリアドレスから, プログラムの構成要素である機械語命令を読み出し, これをデコード (decode) して実行する。機械語命令は, その種類を示す (d) 部と, 演算対象のデータである (e) の格納場所を示すアドレス部から成る。

【選択肢】

(ア) データフロー型 (dataflow architecture)	(イ) オペコード (operation code, opcode)
(ウ) プログラムカウンタ (program counter)	(エ) アーキテクチャ (architecture)
(オ) VLIW (very long instruction word)	(カ) オペランド (operand)
(キ) 命令レジスタ (instruction register)	(ク) メモリデータレジスタ (memory data register)
(ケ) スーパー标カラー (superscalar)	(コ) ノイマン型 (von Neumann architecture)

## 1.3 2015

(2-2) 以下の文章の空欄 (a)～(i) に当てはまる最も適切な語句を, 下記の選択肢から選び, 記号で答えよ。同じ選択肢を複数回用いても良い。

記憶階層の一部を成す仮想記憶 (virtual memory) では, 主記憶装置に置くべきプログラムやデータ (以下では“情報”と呼ぶ) の一部を (a) に置く。これにより, 主記憶装置の持つアドレス空間よりもより広いアドレス空間を有するように, ユーザープログラムに見せかける効果がある。機械語命令 (machine instruction) の中に記載されるアドレスは (b), アドレスバスに送出されるアドレスは (c) と呼ばれる。

仮想記憶において, 必要な情報が主記憶装置上に無く, (d) 上にある場合, この情報は主記憶装置に移動される。この動作は (e) と呼ばれる。この時, 主記憶装置上に空き領域が無ければ, 使用される可能性が低い情報が主記憶装置から (f) へ移動され, 空き領域が作られる。この動作は (g) と呼ばれる。 (e) や (g) が頻繁に発生すると, 計算機の処理性能が低下する。

仮想記憶の代表的な実現方法として, メモリを固定長 (fixed size) のブロック (block) で管理する (h) 方式と, 可変長 (variable size) のブロックで管理する (i) 方式とがある。

【選択肢】

(ア) レジスタ	(イ) スワップイン (swap in)	(ウ) フラグメンテーション (fragmentation)
(エ) キャッシュメモリ	(オ) 仮想アドレス (virtual address)	(カ) セグメンテーション (segmentation)
(キ) 補助記憶装置	(ク) スワップアウト (swap out)	(ケ) コンテキストスイッチ (context switch)
(コ) ページング (paging)	(サ) 実アドレス (real address)	(シ) 直接マッピング (direct mapping)

## 1.4 2014

共有資源 ( shared resource ) の排他制御を実現する方法としてテストアンドセット ( test and set ) やセマフォ ( semaphore ) 等がある。テストアンドセットは一つしかない共有資源の排他制御を実現する機械語命令 ( machine instruction ) である。これは、「共有資源の  a  状態と  b  状態とを監視し制御する」機能を不可分操作 ( atomic operation ) として実行する命令である。不可分操作として実行されるのは以下の二つである。

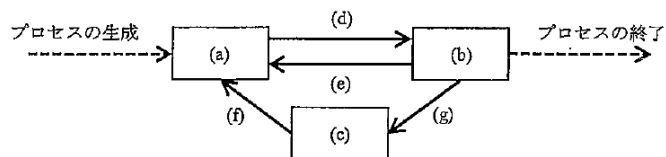
1) テスト：共有資源が  a  状態か  b  状態かをチェックする。  b  状態の場合、終了する。

2) セット：共有資源を  b  状態に変更する。

ここでテストとセットは不可分操作のため、  c  が発生しない。

## 1.5 2013

(2-1) 単一プロセッサを持つマルチプログラミングシステムにおいて、プロセス ( process ) は、実行 ( running ) 状態、実行可能 ( ready ) 状態、待ち ( wait または blocked ) 状態の三つの状態を持ち、プロセスが生成されてから終了するまでその 3 状態間を遷移する。次に示すプロセスの状態遷移図 ( state transition diagram ) の ( a ) ~ ( g ) にあてはまるもっともふさわしい語句を下の選択肢から一つ選び、記号で答えよ。



選択肢

- (ア) システムコール ( system call ) (イ) 入出力処理の完了 (ウ) セグメンテーション ( segmentation )  
(エ) リモートプロシジャコール ( remote procedure call ) (オ) 待ち状態 (カ) ページング ( paging )  
(キ) ディスパッチ ( dispatch ) (ク) 実行状態 (ケ) コンパイル ( compile ) (コ) 横取り ( preemption )  
(サ) 実行可能状態 (シ) バッファリング ( buffering ) (ス) 入出力処理の発生

## 1.6 2012

仮想記憶は、主記憶（メインメモリ; main memory）と二次記憶（secondary storage）（補助メモリ（auxiliary memory）ともいう。）を用いて、計算機で実行されるプロセスに対して、その計算機の主記憶のもつ  よりも大きな  を提供する。代表的な仮想記憶の実現手法としては、主記憶と二次記憶の間のデータ転送を固定長のブロック（block）単位で行う  と、可変長のブロック単位で行う  などがある。

ここで、実アドレス長が 28 [bit]、仮想アドレス長が 32 [bit]、ページサイズを 4 [Kbyte] とする  を導入した計算機について考える。この計算機では、1 [byte] ごとにメモリアドレス（memory address）が割り付けられているとする。また、仮想アドレスから実アドレスへの変換を行う際に用いられるページテーブル（page table）は 1 段で構成されているとし、ページテーブルの 1 エントリは 2 [byte] とする。なお、ページテーブルによる主記憶容量の減少はないものとする。この計算機で利用できる最大の主記憶容量は  [Mbyte] となり、このときページ枠（page frame）は  個となる。一方、仮想アドレス空間の大きさは  [Gbyte] であり、ページテーブルのサイズは  [Mbyte] となる。

選択肢

- (ア) DMA (direct memory access)    (イ) 動的再配置 (dynamic relocation)    (ウ) キャッシュ (cache)  
(エ) コンパクション (compaction)    (オ) 静的再配置 (static relocation)    (カ) ページング (paging)  
(キ) ページ枠 (page frame)    (ク) 実アドレス空間 (real address space)  
(ケ) フラグメンテーション (fragmentation)    (コ) 仮想アドレス空間 (virtual address space)  
(サ) セグメンテーション (segmentation)

## 1.7 2011

プロセッサ高速化技術の一つとしてのパイプライン処理(pipeline processing)は、複数の命令を少しずつずらして同時並行的に実行する方式である。パイプラインを構成するステージでは、以下の処理が行われる。IF ステージでは、[ a ] のアドレスを用いて主記憶から命令語が読み出される。ID ステージでは、命令語の[ b ] 部を解釈し、命令操作が決定される。OF ステージでは、命令語の[ c ] 部によって指定される[ d ] が取り出される。次に、EX ステージでは、命令操作が実行される。MW ステージでは、実行結果の主記憶ないしはレジスタへの[ e ] が行われる。

### 選択肢

- ①スタックポインタ(stack pointer) ②オペランド(operand) ③即値指定(immediate mode)
- ④プログラムカウンタ(program counter) ⑤取り出し(load) ⑥格納(store)
- ⑦アドレス(address) ⑧アドレス変換(address conversion) ⑨分岐命令(branch operation)
- ⑩命令コード(operation code)

## 1.8 2011 2

ファイル(file)は、データやプログラムを格納するための名前付けられた論理的な単位であり、オペレーティングシステム(OS)はファイルシステムを用いてファイルを保存・管理する。OS は、ブロック(block)と呼ばれるデータの最小単位でファイルシステム上のデータ入出力を行う。

ディスク(disk)上のデータ保持の最小単位は[ a ] と呼ばれ、ブロックは連続する1つ以上の[ a ] によって構成される。ディスク上には円盤の同心円を表す[ b ] があり、1つの[ b ] 上に連続する複数の[ a ] が配置されている。一般的にディスクは複数枚並べて構成される場合が多く、その各表面を[ c ] が移動してデータの読み書きを行う。この[ c ] 群を移動することなく読めるディスク面数の[ b ] の組を[ d ] と呼ぶ。ディスク上では「[ d ] 番号、[ c ] 番号、[ a ] 番号」によって、データ格納位置を特定することができる。

### 選択肢

- ①トラック(track) ②バイトオーダー(byte order) ③シリンダ(cylinder)
- ④ヘッド(head) ⑤セクタ(sector) ⑥シーケンス(sequence)

## 1.9 2009

主記憶 (main memory) 内のデータへのアクセスは、語 (ワード, word) を単位とした格納位置を示すアドレス (address) を指定することによって行われる。このとき、アクセスされたデータの写しを小容量であるが高速のキャッシュメモリ (cache memory) (以下、キャッシュ) に一時的に作っておき、次に同じデータがアクセスされたときにキャッシュからデータの写しを取り出すことにより、見かけ上主記憶のアクセス速度を高速化することが広く行われている。これは、一度アクセスされたデータは繰り返し利用される確率が高いという (a) , 近いアドレスに格納されたデータはまとまってアクセスされる確率が高いという (b) を利用するものである。アクセスされるデータの写しがキャッシュに存在する確率をヒット率 (hit ratio) という。

主記憶のデータの写しは、一定の大きさの連続したアドレスに格納されたデータを単位として作られる。このとき、写しを作る大きさで主記憶を先頭から区切ったときの各々の区分をブロック (block) と呼ぶ。また、キャッシュをブロックと同じ大きさに分けた時の各々の区分をブロック枠 (block frame) という。ブロックに対しその写しを作るブロック枠を割り当てることを、ブロックのマッピング (mapping) と呼ぶ。ブロックのマッピングを行うための方式には、任意のブロック枠を割り当てることができる (c) 方式、ブロック枠の割り当てが一意的に決められている (d) 方式がある。(d) 方式は、(c) 方式よりも、キャッシュが同じ容量を持つときのヒット率は一般に (e) く; キャッシュの使用効率も (e) い。また、これら 2 つの方式を組み合わせた方式として、ブロック枠を同数の集合に分け、それぞれの集合に対し (f) 方式を適用する (g) 方式がある。

### 選択肢

- (ア) 空間的局所性 (spatial locality)    (イ) セット連想 (群連想) マッピング (set associative mapping)  
(ウ) 時間的局所性 (temporal locality)    (エ) アドレスマルチプレクス (address multiplex)  
(オ) 直接マッピング (direct mapping)    (カ) 完全連想マッピング (fully associative mapping)  
(キ) ライトスルー (write through)    (ク) ファイルマッピング (file mapping)  
(ケ) 高    (コ) 低



## 1.10 2009 2

単一プロセッサをもつマルチプログラミングシステムにおいて、実行中のプログラムの実体であるプロセスは実行 (running) 状態、実行可能 (ready) 状態、待ち (waiting または blocked) 状態の 3 状態を遷移する。

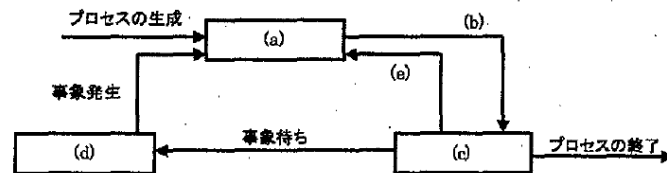


図 1: プロセスの状態遷移

生成されたプロセスはまず (a) となる。その後、そのプロセスは (b) され、(c) に遷移する。(c) にあるプロセスは、入出力処理の完了を待つなどの事象を待つ操作を開始することにより、(d) に遷移する。あるいは、(c) にあるプロセスは、強制的に (e) され、(a) に遷移する。(d) にあるプロセスは、入出力操作の完了や何らかの事象の発生により、(a) に遷移する。このように、生成されたプロセスは 3 状態を遷移し、(c) を経て終了する。なお、実行中のプロセスを次に実行すべきプロセスと切り替えることを (f) という。

### 選択肢

- |                         |                                    |                           |
|-------------------------|------------------------------------|---------------------------|
| (ア) レイテンシ (latency)     | (イ) コンテキストスイッチ (context switch)    | (ウ) 待ち状態                  |
| (エ) コンパイル (compile)     | (オ) ソースプログラム (source program)      | (カ) 実行可能状態                |
| (キ) ディスパッチ (dispatch)   | (ク) 実行可能プログラム (executable program) | (ケ) リンク (link)            |
| (コ) ライブラリ (library)     | (サ) インターセクション (intersection)       | (シ) 実行状態                  |
| (ス) スループット (throughput) | (セ) オブジェクトプログラム (object program)   | (ソ) プリエンプション (preemption) |

## 1.11 2008

図2に示すように、磁気ディスク装置の内部には、両面に(a)の塗られた円盤(ディスク: disk)が複数枚あり、それらが回転軸を共有し、モータ(motor)によって高速回転される。データの記録がディスク上に同心円状になされる場合、その同心円を(b)と言う。(b)は、複数の(c)から構成されており、データは(c)単位で読み書きされる。全てのディスク上の同じ半径を持つ(b)をまとめて(d)と呼ぶ。同じ(d)上に存在するデータは、ディスクが異なってもアクセスする際に(e)を移動させる必要がないため、高速アクセスが可能である。

この装置の性能を決定する指標には、目的の(d)まで(e)が移動するための時間である(f)、ディスクが回転してアクセス対象となる(c)の先頭が(e)のある場所に到達するための時間である(g)、および実際にデータを読み書きするための時間である(h)が用いられる。また、(f)、(g)、(h)の和を一般にアクセス時間と呼ぶ。

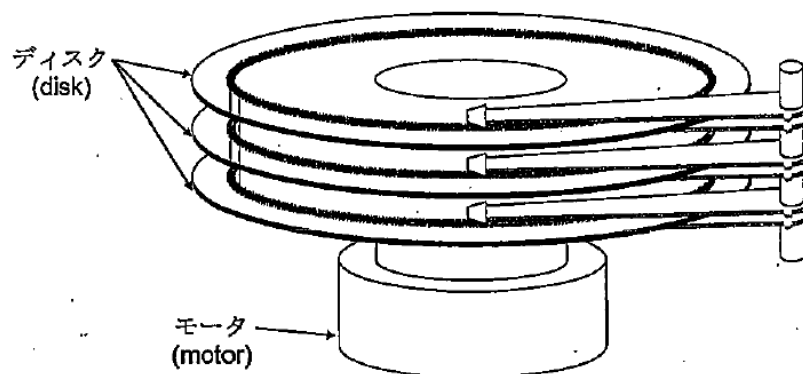


図2: 磁気ディスク装置

### 選択肢

- ①計算時間 (calculation time) ②パーティション (partition) ③セクタ (sector)
- ④転送時間 (transfer time) ⑤ゾーン (zone) ⑥ヘッド (head)
- ⑦サーチ時間あるいは回転遅延時間 (search time) ⑧ボリューム (volume)
- ⑨ディレクトリ (directory) ⑩磁性体 (magnetic material)
- ⑪キャッシュ時間 (cache time) ⑫シリコン (silicon) ⑬トラック (track)
- ⑭シーク時間 (seek time) ⑮シリンダ (cylinder)

## 1.12 2007

単一プロセッサをもつマルチプログラミングシステムにおいて、プロセスは、実行状態(running state)、実行可能状態(ready state)、待ち状態(wait state または blocked state)の3つの状態を持ち、プロセスが生成されてから消滅するまでその3状態間を遷移する。生成されたプロセスは、まず初期状態として (a) となりキュー(queue)につながる。プロセッサに空きが生じると (b) にあるプロセスから1つのプロセスが選択され、その選択されたプロセスはプロセッサに割り付けられ、(c) に遷移する。(d) にあるプロセスは、処理の完了によって消滅する。あるいはプリエンプション(preemption)により、(e) へ遷移しキューにつながる。(f) にあるプロセスが入出力待ちとなった場合には、(g) へ遷移し、入出力が行われた後、(h) へ遷移する。

プロセスをプロセッサに割り付けるスケジューリングアルゴリズムとして、いくつかの方式が考えられる。(i) 方式では、プロセスがキューに到着した順に割り付けられる。(j) 方式では、予想される処理時間の短い順に割り付けられる。これらの方式では、プリエンプションは行われない。(k) 方式では、プロセスがキューに到着した順に割り付けられるが、一定時間(タイムスライス)が経過した後にプリエンプションされ、再びキューの最後につながる。(l) 方式は、新しく到着したプロセスの予想される処理時間とその時点で実行中のプロセスに必要な残りの処理時間を比較し、新しく到着したプロセスの処理時間の方が短ければ、プリエンプションにより新しく到着したプロセスを割り付ける。

### 選択肢

- ① カーネルモード   ② 実行状態   ③ FCFS (First Come Last Served)   ④ ユーザモード
- ⑤ 優先度順   ⑥ FCFS (First Come First Served)   ⑦ 実行可能状態   ⑧ シングルタスク
- ⑨ RR (Round Robin)   ⑩ マルチプロセッサ   ⑪ SRT (Shortest Remaining Time First)
- ⑫ 待ち状態   ⑬ EDF (Earliest Deadline First)   ⑭ SPT (Shortest Processing Time First)

## 1.13 2006

CPU（中央処理装置）は、演算処理部と実行制御部から構成される。図1は、CPUの基本構成の一例を示しており、プログラムカウンタ(PC)、命令レジスタ(IR)、メモリアドレスレジスタ(MAR)、メモリデータレジスタ(MDR)、汎用レジスタ(R1~Rn)、一時的に演算結果などを記憶するレジスタ(X, Y, Z)、算術論理演算器(ALU)が、バスを介して信号を転送している。なお、主記憶装置との信号転送は、アドレスバス、データバスを介して行われる。図中の●は、制御ゲートを示しており、制御信号がオンとなった際にデータを送受できる。cは、桁上げ入力である。

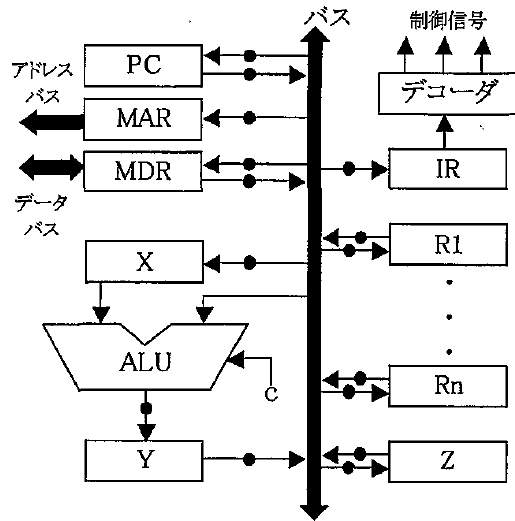


図1 CPUの基本構成

命令フェッチ(命令読み出し)では、(a)の指示する主記憶アドレスから命令を取り出し、(b)に格納する。このことは、次のような複数のステップに分けて実行される。まず、(c)の内容を(d)に転送し、次に主記憶に対して読み出し指示を与える。ここで、主記憶からの読み出し動作が完了し、MDRへデータが格納されるまで、CPUは待機する。なお、この間にPCの内容をALUにより1だけインクリメントしている。MDRへの格納完了を通知する信号を主記憶から受け取った後に、その内容を(e)に転送する。以下の各手順も、同様に複数のステップで実行される。

命令の解釈では、IRに取り出された命令語の命令コード部を解釈し、実行すべき動作を決定する。オペランドフェッチ(オペランド読み出し)では、取り出された命令語の(f)から、オペランドの格納されているレジスタ、あるいは主記憶内の位置を知り、そこからオペランドを取り出す。この際の処理は、(g)で若干異なり、(h)の場合は、命令語のアドレス部で指定された汎用レジスタの内容がオペランドになる。直接アドレス指定の場合は、アドレスそのものが命令語のアドレス部で指定される。(i)の場合は、命令語のアドレス部で指定された汎用レジスタの内容が、実効アドレスとなる。オペランドフェッチが終了すると、命令コード部で指定された演算を実行する。この後、演算結果を指定されたレジスタ、あるいは主記憶に格納する。主記憶に格納するためには、書き込むアドレスを(j)に格納した後、書き込むデータをMDRに格納し、書き込み命令を実行する。引き続き、命令フェッチに戻って、次の命令が実行される。

#### 選択肢

- ① PC ② MDR ③ アドレス部 ④ レジスタ間接指定 ⑤ MAR  
 ⑥ 即値指定 ⑦ インデックス指定 ⑧ IR ⑨ アドレス変換 ⑩ レジスタ  
 ⑪ アドレス指定モード(アドレス指定形式) ⑫ ALU ⑬ レジスタ指定