

ウンコ問題答え（間違ってたらごめん説）

「この答えもウンコです」 せむら

「ウンコも人生には必要なものだから、やまうちよりはマシだよ」 かねじゅん

————アルゴ————

(2013 1)

答え：同等な値が、ソートの最中やソート後でも順序が変わらないソートのこと。

————アーキ————

(2016 2013)

答え：空間的参照局所性 あるデータが参照されると、その近くのデータが参照される可能性が高いこと。あるデータとその近くに位置するデータが同じブロック内に存在することで、同一ブロック内のアクセスが増え、キャッシュのヒット率が上昇するためメモリアクセスの時間が小さくなる。

時間的参照局所性 あるデータが参照されると、近い未来にそのデータが再び参照される可能性が高いこと。最近参照されたデータが含まれるブロックがキャッシュに存在する可能性が高くなり、ヒット率が上昇する。

答え：キャッシュメモリは高速だが小容量であり、メインメモリは低速だが大容量である。両者に階層を作ることで、両者の欠点を補いあい、長所を活かすことができる。

(2015 2)

答え：仮数部のビットが4つごとに無限に循環してしまい、有限桁で値を丸め込む必要があるから。

(2014 2)

答え：減算回路を作る必要がない

答え：テストしてからセットするまでの間で他のプロセスがテストするとヤバイ。

(2013 2)

答え：ブロックサイズが小さい時は、同じブロックに参照したいデータが存在する可能性が低くなり、主記憶への参照が増えるから。

ブロックサイズが大きい時は、キャッシュに保存できるブロックが減るために、ブロック置き換えの回数が増えるから。

(2011 2)

答え：各機能ブロックのうち最も大きいものにする

答え：①ファイルサイズに対してブロックサイズが大きい場合、使用されない領域が増えて効率が悪い。②ブロックサイズが大きいほど同じサイズのファイルにアクセスする場合のシーク回数が減少するため、平均転送時間は短くなる。

(2010)

答え：

条件① 二つ以上のプロセスが同時に危険領域を実行する相互実行を禁止する。

条件② 二つ以上のプロセスが互いに危険領域の実行を無期限に阻止しあうデッドロックを禁止する。

条件③ 危険領域を実行しているプロセスが一つも存在せず、かつ他に要求を出しているプロセスがなければ、危険領域を実行しようとするプロセスにただちに実行許可が与えられる。

(2009)

答え：実装の容易さという観点では **FIFO** が優れている。**LRU** の場合参照した順番を記憶させることが必要となるので実装は難しい。**FIFO** はブロック数を増やしたのにヒット率が下がる減少が発生するので、**LRU** のほうが優れている。

(2008)

答え：エレベータ順ならヘッダが同じ箇所を往復するのは高々二回でありその分アクセスする時間が短くなる。

————— 計算論 —————

(2015 3)

答え：同値関係という言葉が出た時は、反射！対称！推移！って言いながら僕は解答します。

(2013 3)



答え：

文脈自由文法は、正規文法より表現できる言語が多いため、プログラマが使用できる記述方法が多くなる。また、文脈依存文法より表現できる言語を制限しているため、コンパイラがプログラムの構文解析を行う処理が単純になる

(2012 3)

答え：チョムスキー $A \rightarrow BC$ $A \rightarrow \alpha$ ($S \rightarrow \epsilon$)

正則言語 $A \rightarrow \alpha B$ $A \rightarrow \alpha$ 文脈自由言語 $A \rightarrow \text{色々}$

(2011 3(9))

答え：なんか文脈自由言語的なものから、なんだかんだで何個かの解釈ができてしまうと木みたいなものを作るのにそこそこ困る。

○穴埋め解答

(OS 期末)

デマンドページング、プリページング、ハードウェア機構、動的、開始時、コンパイラ機能、静的、メインメモリ

(2016)

エ、コ、ウ、イ、カ

(2015)

キ、オ、サ、キ、イ、キ、ク、コ、カ

(2014)

アンロック、ロック、割り込み

(2013)

サ、ク、オ、キ、コ、イ、ス

(2012)

ク、コ、カ、サ、256、65536、4、2

(2011)

4,12,7,2,6

(2011 2)

5,1,4,3

(2009)

ウ、ア、カ、オ、コ、カ、イ

(2009 2)

カ、キ、シ、ウ、ソ、イ

(2008)

10、13、3、15、6、14、7、4

(2007)

7、7、2、2、7、2、12、7、6、14、9、11

(2006)

1、8、1、5、8、3、11、13、4、5