2013 年 2 月 6 日システムソフトウエア試験問題

学科　　学籍番号＿＿＿＿＿＿＿＿＿＿　　氏名＿＿＿＿＿＿＿＿＿＿

1．　つぎの文章の空白を埋めなさい．（各 1 点：合計 50 点）

(アーキテクチャ:8 点)

- コンピュータは，CPU(中央処理装置)とメモリ，およびその他の入出力装置から構成されている．このうち，CPU は命令の読み取りとデコードなどを行う(a)＿＿＿部と，実際の計算を行う演算部の 2 つから構成されている．(a)部には，演算対象や，読み取った命令と変数などを格納する(b)＿＿＿と呼ばれる小規模で高速なメモリが配置されている．CPU は命令の「読み取り」と「実行」を交互に繰り返すが，最近では一つ前の命令の実行が終わってから次の命令の読み取りをするのではなく，一つ前の命令の実行と次の命令の読み取りを同時に行うことによって高速化を実現する(c)＿＿＿処理が一般的に用いられている．
- CPU が次に読み取る命令の格納番地を指し示す(d)＿＿＿と呼ばれる(b)は，分岐命令などで値が強制的に変更される．
- (e)＿＿＿コールでは，(d)の値をスタックにプッシュしてから分岐先アドレスに変更し，(e)の計算を終えた後にポップすることによって，元の命令実行位置の直後に戻ってくることができる．
- プログラム中の計算結果に応じて(d)の値を書き換える，つまり，実行するプログラムの位置を変えるのが(f)＿＿＿命令である．この命令は，万能チューリングマシンと等価な計算が行えるために不可欠である．
- (f)命令を読み取った場合，条件部の計算結果が確定するまでは，前述の(c)処理で命令を読み取る場所が確定しないため，CPU の計算速度が低下する．(g)＿＿＿文を多用したプログラムの実行速度が低下するのは，このためである．
- この速度低下を防ぐため，(f)が起きる場合と起きない場合の両方の分岐先の命令をともに実行し，分岐先が確定した後に一方を破棄する(h)＿＿＿という機能が備わっているものがある．

(オペレーティングシステム:18)

- 複数のプログラムを短い時間間隔で順次 CPU に割り当てて実行する(i)＿＿＿により，各時刻では単一のプログラムしか実行していないが，ある程度の時間粒度で見ると複数のプログラムが同時に実行されているように見せることができる．
- これら実行中のプログラムは，(j)＿＿＿と呼ばれ，OS からメモリや CPU の割り当ての優先度などの(k)＿＿＿資源の割り当てを受ける．これはさらに(l)＿＿＿と呼ばれる単位に分けられて実行される．
- (l)の実行スケジューリング法として，順番に各(l)の優先順位を上げ，決められたタイムスライス内で実行し，実行した(l)の順位を下げる(m)＿＿＿方式が広く用いられている．
- 複数のプログラムが同時に実行されている(n)＿＿＿環境では，メモリや，補助記憶装置，その他の入出力装置の 1 台に対して同時に複数のアクセス要求が集中す

ることがある．この場合のアクセス要求を調停することも OS の重要な役目である．

- プログラムによるメモリの動的確保と解放は，メモリ領域の断片化を招く．この解消のために，(o) ＿＿＿＿＿＿＿＿ というアドレス変換ハードウエアが用いられる．これは，物理アドレスと (p) ＿＿＿＿＿＿ とを対応づける装置であり，(q) ＿＿＿＿ と呼ばれる固定サイズのメモリを単位として，物理アドレス上の不連続なメモリエリアを論理アドレス上の連続なメモリエリアに対応付ける事が出来る．
- 補助記憶装置の一部を利用して，物理メモリの大きさよりも大きな論理メモリ空間を実現する手法を (r) ＿＿＿＿ と呼ぶ．この実現方式には，(j) が確保している物理メモリ単位で，補助記憶装置に退避・復帰させる (s) ＿＿＿＿＿＿ と，(o) で管理している (q) 単位で補助記憶装置と物理メモリとの間で退避・復帰させる (t) ＿＿＿＿＿＿＿＿ の２種類がある．
- 後者の場合，ある (p) と対応付けられている (q) が退避前と異なる物理アドレスに復帰させられた場合，(o) のアドレス変換により同じ論理アドレスに対応付けられる．
- (t) の効率を上げるために，物理メモリの空き容量が減った時点で補助記憶装置に不要なメモリを退避させておくという処理が必要になる．これを行うのが，(u) ＿＿＿＿＿＿ ・デーモンである．このデーモンの動作を規定するアルゴリズムには，最も古い (q) の置き換え優先度を高くする (v) ＿＿＿＿ ，最も長い期間使われていない (q) の置き換え優先度を高くする (w) ＿＿＿＿ などのアルゴリズムがある．しかし，(v) アルゴリズムでは，物理メモリが多い時の方が補助記憶装置から (q) の内容を読み出す回数が増える (x) ＿＿＿＿ の異常と呼ばれる現象が発生するため，Linux などでは (w) アルゴリズムが用いられる．
- ファイルとは，補助記憶装置上のデータブロックをつなげた構造になっている．このデータブロックのつなげ方は２種類があり，データブロック内に次のデータブロックを指すポインタを書いておく (y) ＿＿＿＿ 構造と，ポインタの一次元並びを用いる (z) ＿＿＿＿＿＿ 構造がある．

(プログラミング言語処理系:24)

- プログラム言語処理系は，C, C++, Pascal, Basic,などの (A) ＿＿ 言語で書かれたプログラムテキストを，(1)CPU が解釈実行可能な (B) ＿＿ 語に変換する (C) ＿＿＿＿ や，(2) 逐次解釈・実行するインタープリタ，などで用いられる．
- プログラムテキストは，(D) ＿＿ 文法で表現可能な字句の集まりとして表現可能であり，プログラムテキストからこれを抽出する字句解析のためにオートマトンが用いられる．
- 字句解析の後には，構文解析が行われる．通常は (E) ＿＿＿＿ 文法を用いてプログラミング言語を定義することができる．
- 構文解析では，(F) ＿＿＿＿ に対してどのような順序で (G) ＿＿＿＿ が適用され，入力として与えられた終端記号の列が生成されたのかを分析する．この分析結果は (H) ＿＿＿ 木で表すことができる．
- (H) 木は，分岐のない中間節をまとめ，演算子を分枝節に引き上げることで (I) ＿＿＿ 木（演算子木とも呼ばれる）に変換することができる．
- 算術式から得られた (I) 木を (J) ＿＿ 走査することにより (K) ＿＿＿＿＿＿ 記法を生成

し，これを入力としてスタック上で式の計算を行うことができる．

- 構文解析の方法は大きく分けて(L)　　型の構文解析と，(M)　　型の構文解析法がある．(L)型構文解析では最左導出を前提とした文法の下での構文解析が行われ，(M)型構文解析では最右導出を前提とした構文解析が行われる．
- (L)型構文解析の例としては(N)　　構文解析，(M)型構文解析の例としては，(O)　　構文解析がある．この方法では，アクション表と GOTO 表をもとにして，スタックを用いて shift と reduce の操作を行いながら構文解析が行われる．このうち reduce は，生成規則の(P)　　辺に対応する部分を見つけて(Q)　　辺に置き換えるという(R)　　操作に対応している．
- 文法と文法にマッチした文字列に対する計算手続きを与えて，プログラミング言語の構文解析器を作るソフトウエアのことを(S)　　と呼ぶ. この例としては, Unix や Linux 上の(T)　　やこれを更に拡張した Bison などがある．これらの(S)で用いられているのは(U)　　という講文解析アルゴリズムである．
- (T)や Bison は単体で用いられることはほとんどなく，字句解析用オートマトンを生成する(V)　　や flex と併用されて，(C)やインタープリタが作成される．
- 言語は文法によって規定された(W)　　記号列の(X)　　集合である．プログラミング言語処理系もある文法に従って作成されており，それを使用するプログラマも文法に従ったプログラムを書いている以上，文法上のエラーは起こさない．その意味で，プログラマが使用する言語の文法を知っておくことは必須である．

２．　下記の TAS 命令と等価な if 文を用いない TAS 命令を右の欄に示しなさい．(15 点)

```
int TestAndSet(int* Vaddr){
        if (*Vaddr) return 1;
        else { *Vaddr=1;
                        return 0;
        }
}
```

```
int TestAndSet(int* Vaddr){



}
```

3.　スレッド間の「優先度逆転」は，何によって生じるか？また，これによってどんな問題が生じるか？さらに，この問題を緩和する 2 つの方法について述べなさい．(10 点)

4. 以下の正規表現（A～D）に対応する記号の集合（すなわち言語）L(A)～L(D)があるとき，L(A)～L(D)の関係はどのようになっているかを BEN 図で表しなさい．但し，集合の交わりの部分がある場合には，その要素も列挙すること．また，a,b,c,d は終端記号を表わす．（15 点）

A:	a*db*
B:	b*ca*
C:	a*cb*
D:	b*da*

5. 下記文法のもとで，「3-2-1」の構文解析を行った結果を，導出木，構文木の形で表わしなさい。（導出木７＋構分木３＝10 点）

<算術式> →<算術式><加減演算子><項>|<項>

<項>→ <項><乗除演算子><因子>|<因子>

<因子>→ <変数>|<数値>|(<算術式>)

<数値>→1|2|... |9

<変数>→A|B|C|...|Z

<加減演算子>→+|－

<乗除演算子>→×|÷

2013 年 2 月 6 日システムソフトウエア試験問題解答例

学科　　学籍番号＿＿＿＿＿＿＿＿＿＿　　氏名＿＿＿＿＿＿＿＿＿＿

1．　つぎの文章の空白を埋めなさい．（各 1 点：合計 50 点）

(アーキテクチャ:8 点)

- コンピュータは，CPU(中央処理装置)とメモリ，およびその他の入出力装置から構成されている．このうち，CPU は命令の読み取りとデコードなどを行う(a)　制御　部と，実際の計算を行う演算部の 2 つから構成されている．(a)部には，演算対象や，読み取った命令と変数などを格納する(b)　レジスタ と呼ばれる小規模で高速なメモリが配置されている．CPU は命令の「読み取り」と「実行」を交互に繰り返すが，最近では一つ前の命令の実行が終わってから次の命令の読み取りをするのではなく，一つ前の命令の実行と次の命令の読み取りを同時に行うことによって高速化を実現する(c)　パイプライン 処理が一般的に用いられている．
- CPU が次に読み取る命令の格納番地を指し示す(d)　プログラムカウンタ　と呼ばれる(b)は，分岐命令などで値が強制的に変更される．
- (e)　サブルーチン　コールでは，(d)の値をスタックにプッシュしてから分岐先アドレスに変更し，(e)の計算を終えた後にポップすることによって，元の命令実行位置の直後に戻ってくることができる．
- プログラム中の計算結果に応じて(d)の値を書き換える，つまり，実行するプログラムの位置を変えるのが(f)　条件分岐　命令である．この命令は，万能チューリングマシンと等価な計算が行えるために不可欠である．
- (f)命令を読み取った場合，条件部の計算結果が確定するまでは，前述の(c)処理で命令を読み取る場所が確定しないため，CPU の計算速度が低下する．(g)　if　文を多用したプログラムの実行速度が低下するのは，このためである．
- この速度低下を防ぐため，(f)が起きる場合と起きない場合の両方の分岐先の命令をともに実行し，分岐先が確定した後に一方を破棄する(h)　投機的実行　という機能が備わっているものがある．

(オペレーティングシステム:18)

- 複数のプログラムを短い時間間隔で順次 CPU に割り当てて実行する(i)　時分割処理　により，各時刻では単一のプログラムしか実行していないが，ある程度の時間粒度で見ると複数のプログラムが同時に実行されているように見せることができる．
- これら実行中のプログラムは，(j)　プロセス　と呼ばれ，OS からメモリや CPU の割り当ての優先度などの(k)計算(機)資源の割り当てを受ける．これはさらに(l)　スレッド と呼ばれる単位に分けられて実行される．
- (l)の実行スケジューリング法として，順番に各(l)の優先順位を上げ，決められたタイムスライス内で実行し，実行した(l)の順位を下げる(m)　ラウンドロビン 方式が広く用いられている．
- 複数のプログラムが同時に実行されている(n)　マルチプログラミング 環境では，メモリや，補助記憶装置，その他の入出力装置の 1 台に対して同時に複数のアクセス要求が集中す

ることがある．この場合のアクセス要求を調停することもOSの重要な役目である．

- プログラムによるメモリの動的確保と解放は，メモリ領域の断片化を招く．この解消のために，(o)　メモリーマネージメントユニット（MMU）というアドレス変換ハードウエアが用いられる．これは，物理アドレスと(p)　論理アドレスとを対応づける装置であり，(q)　ページと呼ばれる固定サイズのメモリを単位として，物理アドレス上の不連続なメモリエリアを論理アドレス上の連続なメモリエリアに対応付ける事が出来る．
- 補助記憶装置の一部を利用して，物理メモリの大きさよりも大きな論理メモリ空間を実現する手法を(r)　仮想記憶と呼ぶ．この実現方式には，(j)が確保している物理メモリ単位で，補助記憶装置に退避・復帰させる(s)　プロセス・スワップと，(o)で管理している(q)単位で補助記憶装置と物理メモリとの間で退避・復帰させる(t)　デマンド・ページングの２種類がある．
- 後者の場合，ある(p)と対応付けられている(q)が退避前と異なる物理アドレスに復帰させられた場合，(o)のアドレス変換により同じ論理アドレスに対応付けられる．
- (t)の効率を上げるために，物理メモリの空き容量が減った時点で補助記憶装置に不要なメモリを退避させておくという処理が必要になる．これを行うのが，(u)　ページアウト・デーモンである．このデーモンの動作を規定するアルゴリズムには，最も古い(q)の置き換え優先度を高くする(v)　FIFO，最も長い期間使われていない(q)の置き換え優先度を高くする(w)　LRUなどのアルゴリズムがある．しかし，(v)アルゴリズムでは，物理メモリが多い時の方が補助記憶装置から(q)の内容を読み出す回数が増える(x)　Bradyの異常と呼ばれる現象が発生するため，Linuxなどでは(w)アルゴリズムが用いられる．
- ファイルとは，補助記憶装置上のデータブロックをつなげた構造になっている．このデータブロックのつなげ方は２種類があり，データブロック内に次のデータブロックを指すポインタを書いておく(y)　リンク構造と，ポインタの一次元並びを用いる(z)　インデックス構造がある．

(プログラミング言語処理系:24)

- プログラム言語処理系は，C, C++, Pascal, Basic,などの(A) 高級言語で書かれたプログラムテキストを，(1)CPUが解釈実行可能な(B) 機械語に変換する(C) コンパイラや，(2)逐次解釈・実行するインタープリタ，などで用いられる．
- プログラムテキストは，(D) 正規文法で表現可能な字句の集まりとして表現可能であり，プログラムテキストからこれを抽出する字句解析のためにオートマトンが用いられる．
- 字句解析の後には，構文解析が行われる．通常は(E) 文脈自由文法を用いてプログラミング言語を定義することができる．
- 構文解析では，(F) 出発記号に対してどのような順序で(G) 生成規則が適用され，入力として与えられた終端記号の列が生成されたのかを分析する．この分析結果は(H) 導出木で表すことができる．
- (H)木は，分岐のない中間節をまとめ，演算子を分枝節に引き上げることで(I) 構文木（演算子木とも呼ばれる）に変換することができる．
- 算術式から得られた(I)木を(J) 後順走査することにより(K) 逆ポーランド記法を生成し，

これを入力としてスタック上で式の計算を行うことができる．

- 構文解析の方法は大きく分けて(L) 下降型の構文解析と，(M) 上昇型の構文解析法がある．(L)型構文解析では最左導出を前提とした文法の下での構文解析が行われ，(M)型構文解析では最右導出を前提とした構文解析が行われる．
- (L)型構文解析の例としては(N) LL 構文解析，(M)型構文解析の例としては，(O) LR 構文解析がある．この方法では，アクション表と GOTO 表をもとにして，スタックを用いて shift と reduce の操作を行いながら構文解析が行われる．このうち reduce は，生成規則の(P)右 辺に対応する部分を見つけて(Q)左 辺に置き換えるという(R)還元 操作に対応している．
- 文法と文法にマッチした文字列に対する計算手続きを与えて，プログラミング言語の構文解析器を作るソフトウエアのことを(S) コンパイラ・コンパイラと呼ぶ. この例としては，Unix や Linux 上の(T) Yacc やこれを更に拡張した Bison などがある．これらの(S)で用いられているのは(U) LALR という講文解析アルゴリズムである．
- (T)や Bison は単体で用いられることはほとんどなく，字句解析用オートマトンを生成する(V) lex や flex と併用されて，(C)やインタープリタが作成される．
- 言語は文法によって規定された(W) 終端 記号列の(X) 無限集合である．プログラミング言語処理系もある文法に従って作成されており，それを使用するプログラマも文法に従ったプログラムを書いている以上，文法上のエラーは起こさない．その意味で，プログラマが使用する言語の文法を知っておくことは必須である．

2． 下記の TAS 命令と等価な if 文を用いない TAS 命令を右の欄に示しなさい．(15 点)

```
int TestAndSet(int* Vaddr){
        if (*Vaddr) return 1;
        else { *Vaddr=1;
                        return 0;
        }
}
```

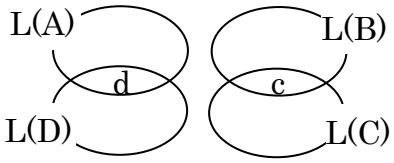
```
int TestAndSet(int* Vaddr){
        int OldValue;
        OldValue = *Vaddr;
        *Vaddr = 1;
        return OldValue;
}
```

3. スレッド間の「優先度逆転」は，何によって生じるか？また，これによってどんな問題が生じるか？さらに，この問題を緩和する 2 つの方法について述べなさい．(10 点)

スレッド間の優先度逆転現象とは，実行優先度が低いスレッド L が，排他制御を必要とする共有資源にアクセスするために，セマフォをロックするなどした際に，それよりも優先度が高いスレッド H が，セマフォのアンロックを待つために停止してしまう現象を指す．(5 点)
この際，セマフォ優先度がスレッド L よりも高く共有資源にアクセスしないスレッド M が存在した場合，L がセマフォをロックしたまま，M の実行が行われ，優先度が高いはずの H が長い間待たされることがあるという問題が生じる．(1 点)
これを回避するには，排他制御の期間中，L を H の優先度で実行する「優先度継承」と，L を最高の優先度で実行する「優先度上限」のいずれかが用いられる．(4 点)

4.　以下の正規表現（A～D）に対応する記号の集合（すなわち言語）L(A)～L(D)があるとき，L(A)～L(D)の関係はどのようになっているかを BEN 図で表しなさい．但し，集合の交わりの部分がある場合には，その要素も列挙すること．また，a,b,c,d は終端記号を表わす．（15 点）

A:	a*db*
B:	b*ca*
C:	a*cb*
D:	b*da*



5.　下記文法のもとで，「3-2-1」の構文解析を行った結果を，導出木，構文木の形で表わしなさい。（導出木 7 ＋構分木 3 ＝10 点）

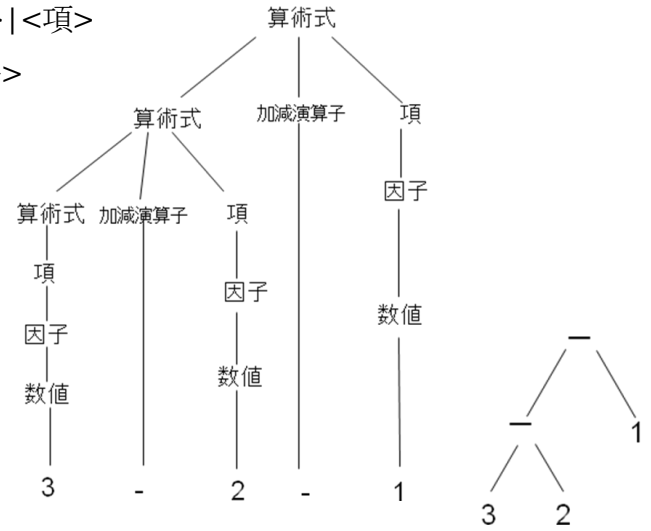
<算術式> →<算術式><加減演算子><項>|<項>

<項>→ <項><乗除演算子><因子>|<因子>

<因子>→ <変数>|<数値>|(<算術式>)

<数値>→1|2|… |9

<変数>→A|B|C|…|Z

<加減演算子>→+|－

<乗除演算子>→×|÷