システムソフトウエア２０１３／１／３０演習問題

学科　　学籍番号＿＿＿＿＿＿＿＿＿＿　　氏名＿＿＿＿＿＿＿＿＿＿

１．　つぎの文章の空白を埋めなさい．

- 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(a)　　とデコードなどを行う制御部と，実際の計算を行う(b)　　部の２つから構成されている．制御部には，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「(a)」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(a)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)　　と呼ばれるレジスタは，分岐命令などで値が強制的に変更される．
- サブルーチンコールでは，(d)の値を(e)　　にプッシュしてから分岐先アドレスに変更し，サブルーチン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．
- 計算結果に応じてプログラムカウンタの値を書き換える，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．この命令を読み取った場合，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，最近の CPU には，条件分岐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)　　的実行という機能が備わっているものがある．
- CPU が周辺機器との間でデータの授受を行う際には，速度差を埋めるために(i)　　を用いてデータを一旦蓄積するバッファリングという処理が用いられる．２つの(i)を交互に用いる(j)　　バッファリング，円状の循環型の(k)　　バッファなどが用いられ，可変個のバッファを動的に確保・解放するためのバッファプールも用いられる．
- バッファリングの際に，FIFO が空になる，あるいは，満杯になるといったイベントが発生し， 次のデータの送出，あるいは，データ送出の中止，等の処理が必要になる場合がある．このようなイベントの検出法には，CPU が機器の変化を繰り返し監視する(l)　　と，電気信号によってイベントの変化を検知する(m)　　の２種類がある．
- (m)の場合は，全てのレジスタを退避させて予め登録しておいた処理ルーチンを実行し，再びレジスタを復帰させて元の処理を継続実行するため，(l)のように CPU が無駄に使われるということはない．
- 現代的な CPU の中には，制御部と演算部の組から成るコアが複数備わった(n)　　，あるいは，制御部のみが二重化された(o)　　と呼ばれる構造のものがある．これらの CPU では同一時刻に異なる命令が同時に実行できる．このような処理を(p)　　処理と呼ぶ．これに対して，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(q)　　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．これを(r)　　処理と呼ぶ．

- 実行中のプログラムは，(s)　　と呼ばれ，OS からメモリやファイル，CPU 割り当ての優先度などの(t)　　の割り当てを受ける．これはさらに(u)　　と呼ばれる単位に分けられて実行される．
- 複数のプログラムが同時に実行されている(v)　　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来ることがある．このような場合に，アクセス要求を調停することも OS の重要な役目である．
- 補助記憶装置に対するアクセス要求の調停は，(w)　　の位置をどう制御するかという問題に帰着する．この問題の最も効率的なアクセスの調停方法は(x)　　アルゴリズムである．
- メモリの動的確保と解放は，メモリ領域の(y)　　を招く．これを防ぐために，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(z)　　とを対応づける装置であり，(A)　　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(B)　　と呼ぶ．この実現方式には，プロセスが確保している物理メモリ単位で，補助記憶装置上のバッキング・ストア領域に退避・復帰させる(C)　　と，MMU で管理している(A)単位でバッキング・ストアと物理メモリとの間で退避・復帰させる(D)　　の 2 種類がある．
- 後者の場合，ある(E)　　アドレスと対応付けられている(A)が退避前と異なる(F)　　アドレスに復帰させられた場合，MMU 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．
- （D）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バッキング・ストア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，(E)　　デーモンである．このアルゴリズムには，最も古いものを退避させる FIFO アルゴリズムや，最も長期間使われていないものを退避させる(F)　　アルゴリズムがある．
- MMU は広大な論理アドレス空間を表現できなければならないが，実際の物理メモリのサイズは，さほど大きくない．このことを利用すると MMU のページテーブルを小さくすることができる．一つは複数の小さいページテーブルを組み合わせた(G)　　であり，もう一つはハッシュ関数を利用した(H)　　である．
- デバイスごとに異なる操作コマンドを open, close, read, write など統一的なインタフェースで扱えるようにするのが(I)　　の役割である．これは，C 言語のプログラムで表現すると，(J)　　を要素とする構造体になる．この構造体が一つのデバイスに対応しており，Unix や Linux の場合は，全デバイスをこの構造体の(K)　　として扱っている．
- ファイルとは，補助記憶装置上の論理的なデータの 1 次元的な並びであり，実際にはセクタ

上のデータブロックを繋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，データブロックを順次繋げていく(L)　　構造と，データブロックを指すポインタの一次元並びを用いる(M)　　構造がある．後者は順アクセスだけでなく(N)　　アクセスにも適合したファイルの構造になっている．

- 小規模な記憶デバイスを複数組み合わせて大規模かつ信頼性の高い記憶デバイスを構築することを RAID : Redundant Array of Independent Disks と言う．これは，大容量補助記憶装置や大容量テープバックアップ装置を作成する際に用いられ，ブロック単位でのパリティ分散記録を行う(O)RAID　　から耐故障性のない(P)　　を行う RAID0 までの様々な方式がある．
- プログラム言語処理系は，C, C++, Pascal, Basic, Fortran，などの(Q)　　言語で書かれたプログラムテキストを，(1)CPU が実行可能な(R)　　語に変換するコンパイラや，(2)逐次解釈・実行する(S)　　，などで用いられる．
- プログラムテキストは，(T)　　文法で表現可能な(U)　　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ためにオートマトンが用いられる．
- 以上に述べた正規表現から字句解析用オートマトンの生成までを行うプログラムとして(V)　　がある．
- 字句解析の後には，構文解析が行われる．通常は(W)　　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，出発記号に対してどのような順序で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(X)　　木として表すことができる．
- この木は，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(Y)　　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(Y)木を後順走査することにより(Z)　　記法　を生成し，これを入力として(e)上で式の計算を行うことができる．

2. なぜ，異なるプロセスに対して，同じユーザアドレス空間が提供されるべきかについて，考えを述べなさい．また，ユーザアドレス空間に対して周辺装置から直接データを DMA 転送で送った場合，どのような問題が発生するかについて，考えを述べなさい．

3. Linux 上の ps コマンドで下記のような結果が得られたとき，２番目の bash を強制的に停止させるためにはどのようなコマンドを実行すれば良いか答えなさい.

UID	PID	PPID	C	STIME	TTY	TIME	CMD
0	2219	2217	0	0:00.37	ttys000	0:01.09	login -pf twada
501	2220	2219	0	0:00.17	ttys000	0:00.21	-bash
0	4577	2220	0	0:00.00	ttys000	0:00.00	ps -fU twada

コマンド：

4. 下記の文法から右下に示すアイテム集合を得た．このとき，アクション表と GOTO 表を示し，衝突が起きている部分を示せ.(右の余白に表を書く．)

(1) <E> →<E>+<B>

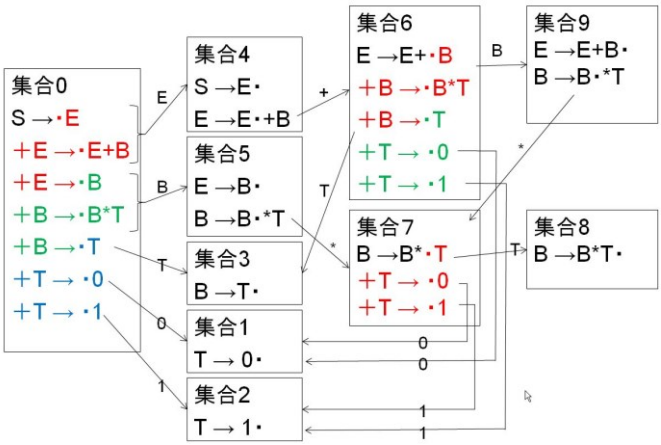
(2) <E> →<B>

(3) <B> →<B>*<T>

(4) <B> →<T>

(5) <T> → 0

(6) <T> → 1



システムソフトウエア２０１３／１／３０演習問題解答例

学科　　学籍番号＿＿＿＿＿＿＿＿＿＿　　氏名＿＿＿＿＿＿＿＿＿＿

１．　つぎの文章の空白を埋めなさい．

- 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．制御部には，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プログラムカウンタと呼ばれるレジスタは，分岐命令などで値が強制的に変更される．
- サブルーチンコールでは，プログラムカウンタ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，サブルーチン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．
- 計算結果に応じてプログラムカウンタの値を書き換える，つまり，実行するプログラムの位置を変えるのが条件分岐命令である．この命令を読み取った場合，前述のパイプライン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．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，最近の CPU には，条件分岐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投機的実行という機能が備わっているものがある．
- CPU が周辺機器との間でデータの授受を行う際には，速度差を埋めるために FIFO を用いてデータを一旦蓄積するバッファリングという処理が用いられる．２つの FIFO を交互に用いるダブルバッファリング，円状の循環型のリングバッファなどが用いられ，可変個のバッファを動的に確保・解放するためのバッファプールも用いられる．
- バッファリングの際に，FIFO が空になる，あるいは，満杯になるといったイベントが発生し， 次のデータの送出，あるいは，データ送出の中止，等の処理が必要になる場合がある．このようなイベントの検出法には，CPU が機器の変化を繰り返し監視するポーリングと，電気信号によってイベントの変化を検知する割り込みの２種類がある．
- 割り込みの場合は，全てのレジスタを退避させて予め登録しておいた処理ルーチンを実行し，再びレジスタを復帰させて元の処理を継続実行するため，ポーリングのように CPU が無駄に使われるということはない．
- 現代的な CPU の中には，制御部と演算部の組から成るコアが複数備わったマルチコア，あるいは，制御部のみが二重化されたハイパースレッディングと呼ばれる構造のものがある．これらの CPU では同一時刻に異なる命令が同時に実行できる．このような処理を並列処理と呼ぶ．これに対して，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．これを並行処理と呼ぶ．

- 実行中のプログラムは，プロセスと呼ばれ，OS からメモリやファイル，CPU 割り当ての優先度などの計算資源の割り当てを受ける．これはさらにスレッドと呼ばれる単位に分けられて実行される．
- 複数のプログラムが同時に実行されている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来ることがある．このような場合に，アクセス要求を調停することも OS の重要な役目である．
- 補助記憶装置に対するアクセス要求の調停は，ヘッドの位置をどう制御するかという問題に帰着する．この問題の最も効率的なアクセスの調停方法は C-Look アルゴリズムである．
- メモリの動的確保と解放は，メモリ領域の断片化を招く．これを防ぐために，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，物理アドレスと論理アドレスとを対応づける装置であり，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仮想記憶と呼ぶ．この実現方式には，プロセスが確保している物理メモリ単位で，補助記憶装置上のバッキング・ストア領域に退避・復帰させるプロセス・スワップと，MMU で管理しているページ単位でバッキング・ストアと物理メモリとの間で退避・復帰させるデマンド・ページングの２種類がある．
- 後者の場合，ある論理アドレスと対応付けられているページが退避前と異なる物理アドレスに復帰させられた場合，MMU 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．
- デマンド・ページング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バッキング・ストア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，ページアウト・デーモンである．このアルゴリズムには，最も古いものを退避させる FIFO アルゴリズムや，最も長期間使われていないものを退避させる LRU アルゴリズムがある．
- MMU は広大な論理アドレス空間を表現できなければならないが，実際の物理メモリのサイズは，さほど大きくない．このことを利用すると MMU のページテーブルを小さくすることができる．一つは複数の小さいページテーブルを組み合わせた多段ページテーブルであり，もう一つはハッシュ関数を利用した逆ページテーブルである．
- デバイスごとに異なる操作コマンドを open, close, read, write など統一的なインタフェースで扱えるようにするのがデバイスドライバの役割である．これは，C 言語のプログラムで表現すると，関数のポインタを要素とする構造体になる．この構造体が一つのデバイスに対応しており，Unix や Linux の場合は，全デバイスをこの構造体の配列として扱っている．
- ファイルとは，補助記憶装置上の論理的なデータの１次元的な並びであり，実際にはセクタ上のデータブロックを繋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，データブロックを順次繋げていくリンク構造と，データブロックを指すポインタの一次元並びを用いるインデックス構造がある．後者は順アクセスだけでなくランダムアクセス

にも適合したファイルの構造になっている．

- 小規模な記憶デバイスを複数組み合わせて大規模かつ信頼性の高い記憶デバイスを構築することを RAID：Redundant Array of Independent Disks と言う．これは，大容量補助記憶装置や大容量テープバックアップ装置を作成する際に用いられ，ブロック単位でのパリティ分散記録を行う RAID5 から耐故障性のないストライピングを行う RAID0 までの様々な方式がある．
- プログラム言語処理系は，C, C++, Pascal, Basic, Fortran，などの 高級 言語で書かれたプログラムテキストを，(1)CPU が実行可能な 機械 語に変換するコンパイラや，(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，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ためにオートマトンが用いられる．
- 以上に述べた正規表現から字句解析用オートマトンの生成までを行うプログラムとして lex がある．
- 字句解析の後には，構文解析が行われる．通常は 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，出発記号に対してどのような順序で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 導出 木として表すことができる．
- この木は，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 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 構文 木を後順走査することにより 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下降型の構文解析と，上昇型の構文解析法がある．下降型構文解析では 最左 導出を前提とした文法の下での構文解析が行われ，上昇型構文解析では 最右 導出を前提とした構文解析が行われる．
- 下降型構文解析の例としては LL 構文解析や，これと本質的に等価な再帰下降型構文解析が挙げられる．
- 上昇型構文解析の一例としては， 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．reduce は，生成規則の 右 辺に対応する部分を見つけて 左 辺に置き換えるという 還元 操作に対応している．

2. なぜ，異なるプロセスに対して，同じユーザアドレス空間が提供されるべきかについて，考えを述べなさい．また，ユーザアドレス空間に対して周辺装置から直接データを DMA 転送で送った場合，どのような問題が発生するかについて，考えを述べなさい．

複数のプログラムが同じ絶対番地を参照する場合などには，プログラムを再配置したり，参照する番地を書き換えたりしなければならなくなる．これを回避するために個々のプロセスに対してユーザアドレス空間が用意されるようになった．

ユーザアドレス空間は，MMU によって実際の物理アドレスに対応づけられており，仮想記憶機構の働き（ページング）により，この対応関係が時間とともに変化してしまう．この結果，外部デバイスから DMA で物理アドレスに転送されたデータが，対応関係が変化した結果，ユーザアドレス空間から参照できなくなるという現象が起こりうる．

3. Linux 上の ps コマンドで下記のような結果が得られたとき，2 番目の bash を強制的に停止させるためにはどのようなコマンドを実行すれば良いか答えなさい．

UID	PID	PPID	C	STIME TTY	TIME CMD
0	2219	2217	0	0:00.37 ttys000	0:01.09 login -pf twada
501	2220	2219	0	0:00.17 ttys000	0:00.21 -bash
0	4577	2220	0	0:00.00 ttys000	0:00.00 ps -fU twada

kill –STOP 2220

4. 下記の文法から右下に示すアイテム集合を得た．このとき，アクション表と GOTO 表を示し，衝突が起きている部分を示せ

(1) <E> →<E>+<B>

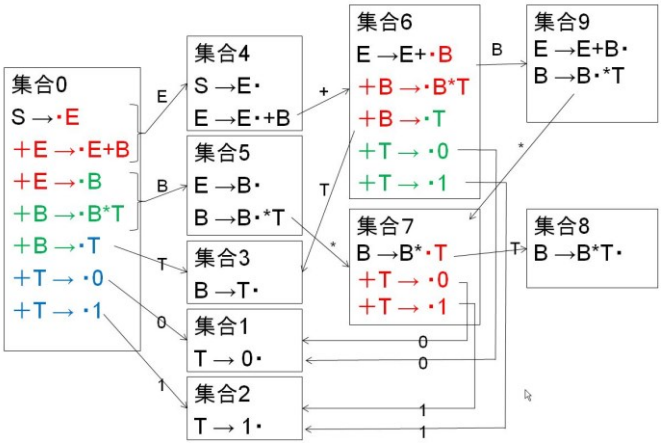
(2) <E> →<B>

(3) <B> →<B>*<T>

(4) <B> →<T>

(5) <T> → 0

(6) <T> → 1



	アクション						GOTO		
状態	*	+	0	1	$		E	B	T
0			s1	s2			4	5	3
1	r5	r5	r5	r5	r5				
2	r6	r6	r6	r6	r6				
3	r4	r4	r4	r4	r4				
4		s6			acc				
5	s7 r2	r2	r2	r2	r2				
6			s1	s2				9	3
7			s1	s2					8
8	r3	r3	r3	r3	r3				
9	s7 r1	r1	r1	r1	r1				