

3104 市川 敬士

一章

1. オペレーティングシステムの目的を簡単に述べよ

- ユーザーと計算機ハードウェアの間であって、ユーザーに代わって計算機システムの資源を効率よく管理し、ユーザーに対して使いやすい環境を提供する

2. 計算機のハードウェア資源の例を3つ述べよ

- キーボード
- CPU
- メモリ

3. OSのサービスは使用環境や目的によって異なる。その例をあげよ。

- プログラム開発に使用されるOSと制御装置に使用されるOSについて比較を行う
 - プログラム開発
 - 会話型の処理に向けたサービスが必要となる。
 - 制御装置
 - 外部からの事象に高速に対応するためのサービスが必要となる。

4. モジュールの情報隠蔽の目的を述べよ。

- あるモジュールを変更したことによって他のモジュールに変更を及ぼす機会を少なくするため。

5. 抽象データ型について述べよ

- プログラミング言語で扱うデータ型の一種で、基本的なデータ型の変数を組み合わせたデータ構造と、そのデータ構造に対する操作を一つのまとまりとして定義したもの。

6. 方針と機構の分離の目的を答えよ

- 各々の部分の変更や拡張を独立して行えるようにするため。

7. 階層化の利点を述べよ

- 上下以外のモジュールへ与える影響を少なくできる。

8. モノシリックカーネルとマイクロカーネルについて説明せよ

- モノシリックカーネル
 - すべてのサービスを取り込んだカーネル
- マイクロカーネル
 - サービスの必要最低限のみを持つカーネルであり、基本サービス以外はシステムサーバーが担当している。

9. OSのインストールの流れを述べよ

1. OSが入っているメディア(DVD-ROMなど)にあるインストーラを起動

2. インストーラがメディアの中のOS本体を2時記憶へコピー
3. ハードウェアに関するパラメータを設定
4. OS本体に関するパラメータを設定
5. その他のプログラムのコピー・設定

10. OSが起動するまでの流れを述べよ、但し、"ブーストラップ"と"ローダ"の2語を含むこと

- 電源起動後、まずブーストROMにあるブーストラップが起動する。
このブーストラップは2次記憶にあるOSのローダを主記憶へ読み込む。
すると、ローダが起動し、2次記憶中のOS本体を主記憶に読み込む。
最後にOSが起動し、各種初期化後サービスを開始する。

11. OSの稼働情報として採取される事柄を書け。また、稼働情報を採取する目的を述べよ。

- 採取される事柄
 - システム統計情報
 - システムログ
 - システム動作情報
- 採取する目的
 - 採取した情報を解析することによって、よりシステムを円滑に運用するため。

2章

1. プロセスとは何かを簡単に述べよ。

- 特定の仕事を遂行するための一連の捜査系列を実行する活動のこと。

2. プロセスの状態遷移図を描け。

- pngファイル

3. 実行状態・実行可能状態・待ち状態について説明せよ。

- 実行状態
 - CPUが割り当てられている状態のこと。
- 実行可能状態
 - CPUが割り当てられるのを待機している状態のこと。
- 待ち状態
 - 事象を待っている状態のこと。

4. PCBとは何か。

- プロセス制御ブロックといい、プロセスに関する情報の集合を意味する。その集合には以下のものが含まれる。
 - プロセス識別子
 - プロセスの状態
 - プロセスの優先度
 - 各種メモリ領域へのポインタ
 - レジスタの退避領域
 - プログラムカウンタ

- etc...

5. 文脈切り替えの仕組みを説明せよ。

- CPUが処理するプロセスを切り替える、つまり、参照するPCBを切り替えることを意味する。

6. unixでのfork-execモデルについて説明せよ

- fork
 - 親プロセスをコピーして新しいプロセス(子プロセス)を作成
- exec
 - 子プロセスのメモリ空間を新しいプログラムで置き換える
- 以上の方法でより、親プロセスと子プロセスを並行して動作することのできるプロセス生成と実行の手法のことを指し示す

7. 再入可能プログラムと再使用不能プログラムの違いについて述べよ。

- 再入可能
 - プロセスごとにデータ領域が独立しており、自分自身でプログラムを書き換えず、同時に複数のプログラムを生成し、実行できること。
- 再使用不能
 - 再入ができず、複数のプログラムを生成した場合、正しい実行が行えない。また、プロセスが終了した時、再実行できるならば、それは逐次再使用可能と呼ばれる。

8. プロセスに比べて、スレッドが有利な点を述べよ。

- オーバーヘッドが小さく、コード領域とデータ領域を共有できるため、同じ目的の処理の並列化やCPU処理と入出力処理の並列化を行うことができる点。

9. スレッドの特長を活かした応用例をあげよ

- サーバプログラムの応用性の向上

10. マルチプログラミングの目的について述べよ

- CPUのアイドル時間を減らし、CPUの使用率を向上させること。

11. CPUバウンドプロセスと入出力バウンドプロセスの違いについて述べよ

- CPUバウンドプロセス:CPU処理の割合が入出力処理よりも多いプロセスのこと。
- 入出力バウンドプロセス:CPU処理の割合が入出力処理よりも少ないプロセスのこと。

12. CPUスケジューラとディスパッチャの役割について述べよ

- CPUスケジューラ:実行可能状態にあるプロセスの集合から次にCPUを割り付けるプロセスを選択すること。
- ディスパッチャ:実際にCPUを割り付けること。

13. スケジューリングの評価基準を3つ以上あげよ

- マルチプログラミングの多重度:システム中に存在するプロセス数
- CPU使用率:プロセスが終了するまでにCPUを使用していた割合
- スループット:単位時間あたりに終了するプロセスの数

- ターンアラウンド時間:プロセスを生成してから終了するまでの時間
- 待ち時間:プロセスが終了するまでに実行可能状態であった時間
- 応答時間:プロセスが生成されてから、最初に処理されるまでの時間

14. CPU使用率・スループット・ターンアラウンド時間・待ち時間・応答時間について説明せよ

- CPU使用率:プロセスが終了するまでにCPUを使用していた割合
- スループット:単位時間あたりに終了するプロセスの数
- ターンアラウンド時間:プロセスを生成してから終了するまでの時間
- 待ち時間:プロセスが終了するまでに実行可能状態であった時間
- 応答時間:プロセスが生成されてから、最初に処理されるまでの時間

15. FCFSスケジューリング・SJFスケジューリング・優先度スケジューリング・ラウンドロビンスケジューリング・多重レベルスケジューリングについてそれぞれ簡単に説明せよ。

- FCFSスケジューリング:はじめにきたプログラムから順にCPUを割り当てていく方法のこと
- SJFスケジューリング:処理時間の短い順にCPUを割り付ける方法のこと
- 優先度スケジューリング:優先度が高い順にCPUを割り付けていく方法のこと
- ラウンドロビンスケジューリング:ある一定の時間間隔ごとに強制的に割り当てるCPUをかえていく方法のこと。
- 多重レベルスケジューリング:プロセスの特性に合わせて、スケジューリングアルゴリズムやパラメータを使い分ける方法のこと

16. 無限の延期とエージングについて説明せよ。

- 無限の延期:優先度の低いプロセスが優先度の高いプロセスのためにいつまでもCPUを割り付けられないこと
- エージング:システムに長い時間滞在しているプロセスの優先度を徐々に高くすること。無限の延期に対する対策のひとつ。

17. 教科書p40問2.10を解け。

- pngファイル

従って、

- FCFS
 - 平均ターンアラウンド時間:13
 - 平均待ち時間:10
- SJF
 - 平均ターンアラウンド時間:7
 - 平均待ち時間:3
- ラウンドロビン
 - 平均ターンアラウンド時間:10
 - 平均待ち時間:6
- 優先度
 - 平均ターンアラウンド時間:9
 - 平均待ち時間:5

1. 平行プロセス、逐次プロセスの違いを説明せよ

- 平行プロセスは同時に複数実行可能なプロセスのことを指すが、逐次プロセスは同時に複数実行不可能なプロセスのことを指す。

2. 共同型逐次プロセスが同期を取る必要があるのは何故か

- 同時に使用することによる矛盾を回避するために高々1つのプロセスが資源を占有できる様にするため

3. 平行プロセスが素であるとはどういうことか。また、交差しているとはどういうことか

- 素である
 - 共通に操作するプロセスが存在しない
- 交差している
 - 共通のデータを操作する平行プロセスのこと

4. 相互排除を実現するためには、不可分な操作が必要である。TS命令を例にして不可分な操作の流れを述べよ。

- 不可分な操作は
 - 操作を開始し、
 - 操作を実行し、
 - 操作を終了させる。
- が一連の流れとなっている。
TS命令ではそれぞれ
 - 共有変数lockに対してTS命令を呼び出し、
 - lock変数の値を読み取り、新しい値を設定し、
 - 元の値を返す
- といった流れで行われる。

5. 共有資源・逐次的資源について説明せよ

- 共有資源
 - プロセス間で共有されるプログラムやデータなどの資源のこと
- 逐次的資源
 - 共有資源であっても同時に1つのプロセスしか使用できない資源のこと

6. 臨界領域とは何か

- 各々のプロセスにおいて、逐次的資源を使用する部分のこと

7. Dekkerのアルゴリズムを説明せよ

- 最も低水準の相互排除操作は主記憶への格納操作が不可分であることを利用すること

8. TS命令を説明せよ

- test and set命令の略であり、以下の2つを同時に行う単一の機械命令のこと
 - 共有変数の値の読み出しとテスト
 - 共有変数への新しい値の設定

9. セマフォはTS命令の問題点をどのように改善しているか

- セマフォは低レベルの動機機構を使用することの問題を指摘し、それを解決するための1つの策である。
これは、

10. セマフォのP操作とV操作について説明せよ

- P操作
 - 1つの引数を必要とし、ここではそれをS仮定する。P(s)が実行されれば $S > 0$ となるまで待機し、そうなった時に $s := s - 1$ を実行する。
- V操作
 - 1つの引数を必要とし、その値に1加算する。

11. モニタはセマフォのどのような問題点を改善できるか

- モニタは共有資源とそれに対する操作(確保、解放など)、初期状態のためのコードが一体化した構造体を持つ。従って共有資源に対する動機基本命令はそれを操作するモニタ内にのみ指定されるため、指定の誤りがモニタ内に局所化されること。