VO 182.022 28. Jänner 2008

Prüfung Betriebssysteme

KNr. MNr. Zuname, Vorname

Ges.)(100) 1.)(25) 2.)(31) 3.)(20) 4.)(24) Zusatzblätter:

Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!

1 File Management (25)

1.1 Allokation (6)

Welche Arten der Allokation kennen Sie zur Implementierung einer Datei im sekundären Speicher? Erklären Sie diese und erläutern Sie Vor- und Nachteile!

1.2 File Management Operationen (4)

Nennen und erklären Sie vier typische Operationen beim File Management!

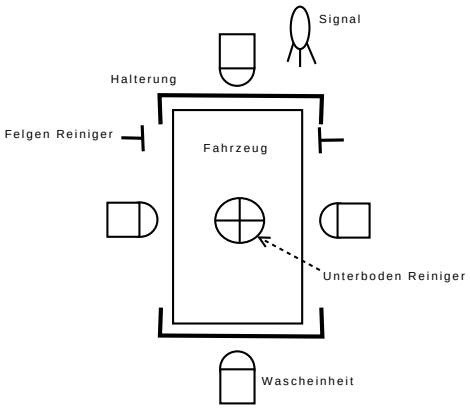
1.3 File Management Operationen (10)

Nennen und erklären Sie fünf fundamentale Arten der Datei-Organisation!

1.4 Block-Verwaltung (5)

Welche Arten der Block-Verwaltung kennen Sie? Erklären Sie ein Verfahren der Block-Verwaltung anhand eines Beispiels!

2 Synchronization (31)



Synchronisieren Sie die Tätigkeiten der Komponenten einer Waschanlage während einer Autowäsche in der Waschbox. Ein Waschvorgang läuft folgendermaßen ab:

- Der Fahrer fährt in die Waschbox (`enter_box()`). Ein Signalgeber gibt mittels eines Lichtsignals, welches auf 'Rot/Red' steht, die Anweisung das Fahrzeug in der Box zum Stillstand zu bringen (`brake()`). Der Fahrer darf erst wieder Gas geben (`accelerate()`), wenn die Waschеinheiten fertig sind und das Lichtsignal auf 'Grün/Green' gestellt ist. Damit ist der Waschvorgang beendet.

- Das Lichtsignal signalisiert dem Fahrer mit der Anzeige 'Rot/Red' bzw. 'Grün/Green' (`signal_red()` und `signal_green()`), ab wann das Fahrzeug wieder bereit ist, die Waschbox zu verlassen. Das Lichtsignal stellt sich auf 'Grün/Green' um, sobald der Waschvorgang beendet ist und das Fahrzeug nicht mehr fixiert ist.

- Um den Waschvorgang zu ermöglichen, muss das Auto in der Waschbox fixiert werden. Diese Aufgabe übernehmen zwei Halterungen, sobald das Auto still steht und das Lichtsignal dies mittels 'Rot/Red' anzeigt. Eine Halterung fixiert das Auto vorne (`lock_front()`) und eine weitere hinten (`lock_back()`). Erst dann können die Waschеinheiten, die Felgenreiniger und die Unterbodenwaschеinheit mit dem Waschvorgang beginnen. Sobald diese fertig sind und dies melden, kann das Fahrzeug wieder von der Haltevorrichtung befreit werden (`unlock_front()` und `unlock_back()`) damit es die Waschbox verlassen kann.

- Insgesamt stehen 4 Waschеinheiten bereit für die Fahrzeugreinigung, je eine für jede Seite (vorne, hinten, rechts, links). Jede Waschеinheit besteht aus 4 Funktionen: Eine Düse sprüht Wasser (`sprinkle_water()`) bzw. verteilt Luft nach dem Waschvorgang (`sprinkle_air()`) und gibt ein Signal sobald der Waschvorgang beendet ist (`ready()`). Eine andere Düse verteilt Waschmittel (`sprinkle_detergent()`). Nach dem Versprühen von Wasser und dem Einbringen von Waschmittel werden die Waschbürsten bewegt (`activate_brushes()`).

- Zwei Waschеinheiten sind für die Reinigung der Felgen (engl.: Rim) zuständig (`clean_rim()`). Sobald die Felgen gereinigt sind, wird das mittels eines Signals gemel-

det (`ready()`). Damit kann das Fahrzeug seitens der Felgenreinigungseinheit wieder aus der Halterung gelöst werden.

- Eine Unterbodenreinigungseinheit säubert den Unterboden des Fahrzeugs (`clean_underbody()`). Sobald der Unterboden gereinigt ist, wird ein Signal gegeben (`ready()`). Damit kann das Fahrzeug von Seiten der Unterbodenreinigungseinheit freigegeben werden.

Ergänzen Sie den folgenden Code entsprechend. Beachten Sie dabei folgende Punkte

- Die Synchronisation hat durch Semaphore zu erfolgen, welche in der Funktion `Init()` zu initialisieren sind. Dafür steht die Funktion `initsem(semaphor,value)` zur Verfügung. Zur Synchronisation sind `P(semaphor)` und `V(semaphor)` zu verwenden. Verwenden Sie eine minimale Anzahl von Ressourcen!
- Achten Sie auf maximale Parallelität, um den Boxenstopp so schnell wie möglich zu beenden!

Initialisierung:

```
Init() {
```

```
}
```

Der Fahrer:

```
Driver() {
```

```
}
```

Wascheinheiten:

Obwohl diese Funktion viermal aufgerufen wird (für jede Seite), braucht diese Funktion nur einmal programmiert werden. Diese Funktion implementiert die Tätigkeit von je drei Wascheinheiten pro Fahrzeugseite.

```
Cleaning_Units() {
```



```
}
```

Halterungen zur Fixierung des Fahrzeugs:

Diese Funktion implementiert die Tätigkeit der zwei Halterungen, welche das Auto fixieren (vorne und hinten).

```
Lock_Car() {
```

```
}
```

Die Unterbodenreinigungseinheit:

```
Underbody_Unit() {
```



```
}
```

Reinigung der Felgen:

```
Rim_Unit() {
```



```
}
```

Das Signallicht:

```
Signal() {
```



```
}
```

3 Security (20)

3.1 Begriffe (8)

Was versteht man unter *Confidentiality (Secrecy)*, *Integrity* und *Availability*? Erklären Sie die drei Begriffe.

- Confidentiality:

- Integrity:

- Availability:

Arten der Bedrohung (Types of Threats)

Füllen Sie folgende Tabelle derart aus, dass Sie jede angegebene Art der Bedrohung einem der Begriffe *Confidentiality (Secrecy)*, *Integrity* und *Availability* zuordnen (Fehlende Antworten werden negativ, falsche Antworten werden doppelt negativ gewertet!):

Art der Bedrohung	**bedroht**
Interruption	
Interception	
Modification	
Fabrication	

3.2 Design Principles for Security (8)

Nennen Sie mindestens 4 Design Prinzipien für Security und beschreiben Sie diese kurz:

3.3 Maßnahmen gegen Passwortattacken (4)

Nennen Sie mindestens 4 Maßnahmen gegen Passwortattacken:

4 Deadlock (24)

Bedingungen für Deadlock (8)

Erklären Sie die für das Auftreten eines Deadlocks notwendigen und hinreichenden Bedingungen.

Behandlung von Deadlocks (8)

Erklären Sie den Begriff der Deadlock Prevention. Führen Sie zu jedem der beiden Arten von Deadlock Prevention ein Beispiel für eine Strategie zur Behandlung von Deadlocks an.

Worin liegt der Unterschied zwischen Deadlock Prevention und Deadlock Avoidance?

Process Initiation Denial (8)

Gegeben ist ein System von *3* Prozessen (P1, P2 und P3) und deren Geamtanforderung an Ressourcen, dargestellt durch die *Claim-Matrix C*. Die Prozesse verwenden *3* Ressourcenkategorien (R1, R2 und R3), deren insgesamt zur Verfügung stehende Gesamtanzahl durch den Vektor R abgebildet wird.

Im aktuellen Zustand des Systems werden die Prozesse P1 und P3 bereits ausgeführt. Bescreiben Sie anhand dieses Systemzustandes die gegenwärtige Ressourcenbelegung durch Ausfüllen der *Allocation-Matrix A* und des *Available-Vektors V* (in den Matrizen repräsentiert jede Spalte eine Ressource und jede Zeile einen Prozess).

$$R = \begin{pmatrix} 80 & 5 & 6 \end{pmatrix} \quad V = \begin{pmatrix} & & \end{pmatrix}$$

$$C = \begin{pmatrix} 10 & 1 & 2 \\ 40 & 1 & 4 \\ 16 & 2 & 2 \end{pmatrix} \quad A = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

Darf mit Programm 2 gemäß *Process Initiation Denial* begonnen werden?

◯ Ja ◯ Nein

Begründen Sie Ihre Antwort (Antworten ohne Begründung werden nicht gewertet!):