

Aufgabe 1: (30 Punkte)

Bei den Multiple-Choice-Fragen ist jeweils nur **eine** richtige Antwort eindeutig anzukreuzen. Auf die richtige Antwort gibt es die angegebene Punktzahl.

Wollen Sie eine Multiple-Choice-Antwort korrigieren, kreisen sie bitte die falsche Antwort ein und kreuzen die richtige an.

Lesen Sie die Frage genau, bevor Sie antworten.

- a) Beim Einsatz von RAID 5 wird durch eine zusätzliche Festplatte Datensicherheit erzielt, so dass der Ausfall einer Festplatte den laufenden Betrieb nicht stören kann. Welche Aussage dazu ist richtig? (1 Punkt) 2 Punkte
- Es sind mindestens 5 Festplatten nötig.
 - Die Paritätsinformation wird gleichmäßig über alle beteiligten Platten verteilt.
 - Der Lesezugriff auf ein Plattensystem mit RAID 5 ist langsamer als bei normalen Plattenzugriffen, da der Zugriff auf die Platten komplexer ist.
 - Es dürfen nicht mehr als 5 Festplatten beteiligt sein, da sonst die Paritätsinformation nicht mehr gebildet werden kann.
- b) Man unterscheidet Traps und Interrupts. Welche Aussage ist **richtig**? 3 Punkte
- Bei der mehrfachen Ausführung eines unveränderten Programms mit gleicher Eingabe treten Interrupts immer an den gleichen Stellen auf.
 - Der Zeitgeber (Systemuhr) unterbricht die Programmbearbeitung in regelmäßigen Abständen. Die genaue Stelle der Unterbrechungen ist damit vorhersagbar. Somit sind solche Unterbrechungen in die Kategorie Trap einzuordnen.
 - Der Zugriff auf eine logische Adresse kann zu einem Trap führen.
 - Wenn ein Interrupt einen schwerwiegenden Fehler signalisiert, muss das unterbrochene Programm abgebrochen werden.

- c) Welche der folgenden Aussagen zu statischem bzw. dynamischem Binden ist **falsch**? 2 Punkte
- bei dynamischem Binden werden alle Adressen bis zum Zeitpunkt des Programmstarts aufgelöst
 - bei dynamischem Binden können auch zum Übersetzungszeitpunkt alle bekannten Adressbezüge bereits vollständig aufgelöst werden
 - beim statischen Binden werden alle Adressen zum Bindezeitpunkt aufgelöst
 - statisch gebundene Programme können zum Ladezeitpunkt an beliebige Speicheradressen platziert werden
- d) Welche der folgenden Aussagen bzgl. Threads ist **falsch**? 2 Punkte
- Die Einlastung (das Dispatching) eines Threads ist eine privilegierte Operation und kann deshalb grundsätzlich immer nur durch das Betriebssystem vorgenommen werden.
 - Das Betriebssystem ist nicht in der Lage, einen einzelnen User-Level Thread bei einer fehlerhaften Operation (z. B. Segmentation fault) gezielt abzubrechen.
 - Ein Anwendungsprogrammierer kann die Schedulingstrategie für seine User-Level Threads selbst programmieren.
 - Die Erzeugung eines Kernel-Level Threads ist teurer als die Erzeugung eines User-Level Threads.
- e) Bei einer prioritätengesteuerten Prozess-Auswahlstrategie (Scheduling-Strategie) kann es zu Problemen kommen. Welches der folgenden Probleme kann auftreten? 2 Punkte
- Die Anzahl der Prioritäten reicht nicht aus, wenn nur wenige Prozesse vorhanden sind.
 - Prozesse können ausgehungert werden.
 - Das Phänomen der Prioritätsumkehr hungert niedrig-priore Prozesse aus.
 - Die Auswahlstrategie arbeitet ineffizient, wenn viele Prozesse im Zustand bereit sind.

- f) Welche Problematik wird durch das Philosophenproblem beschrieben? 2 Punkte
- Ein Erzeuger und ein Verbraucher greifen gleichzeitig auf gemeinsame Datenstrukturen zu.
 - Mehrere Prozesse greifen lesend und schreibend auf gemeinsame Datenstrukturen zu.
 - Exklusive Bearbeitung durch mehrere Bearbeitungsstationen.
 - Gleichzeitiges Belegen mehrerer Betriebsmittel.
- g) Im regulären Ablauf eines Anwendungsprogramms und einer Signalbehandlungsfunktion (bzw. im Ablauf eines Betriebssystems und einer Unterbrechungsbehandlungsfunktion) wird auf gemeinsame Daten modifizierend zugegriffen. Welche Aussage ist falsch? 3 Punkte
- Alle Verfahren der mehrseitigen Synchronisation können eingesetzt werden
 - Der Aufruf von P-Operationen im Anwendungsprogramm und dazu korrespondierenden V-Operationen in der Signalbehandlung führt nicht zu Verklemmungen
 - Falscher Einsatz von Schlossvariablen (lock/unlock-Operationen) kann zu Verklemmungen führen
 - Alle Verfahren nicht-blockierender Synchronisation sind einsetzbar
- h) Virtualisierung kann als Maßnahme gegen Verklemmungen genutzt werden. Warum? 2 Punkte
- Im Fall einer Verklemmung können zusätzliche virtuelle Betriebsmittel neu erzeugt werden. Diese können dann eingesetzt werden, um die fehlenden physikalischen Betriebsmittel zu ersetzen.
 - Durch Virtualisierung kann man über Abbildungsvorgänge Zyklen, die auf der logischen Ebene vorhanden sind, auf der physikalischen Ebene auflösen.
 - Eine Verklemmungsauflösung ist einfacher, weil virtuelle Betriebsmittel jederzeit ohne Schaden entzogen werden können.
 - Durch Virtualisierung ist ein Entzug von physikalischen Betriebsmitteln möglich, obwohl dies auf der logischen Ebene unmöglich ist.

- i) Wie funktioniert Adressraumschutz durch Eingrenzung? 2 Punkte
- Der Lader positioniert Programme immer so im Arbeitsspeicher, dass unerlaubte Adressen mit nicht-existierenden physikalischen Speicherbereichen zusammenfallen.
 - Begrenzungsregister legen einen Adressbereich im logischen Adressraum fest, auf den alle Speicherzugriffe beschränkt werden.
 - Begrenzungsregister legen einen Adressbereich im physikalischen Adressraum fest, auf den alle Speicherzugriffe beschränkt werden.
 - Die MMU kennt die Länge eines Segments und verhindert Speicherzugriffe darüber hinaus.
- j) Welche Aussage zu Speicherzuteilungsverfahren ist **falsch**? 2 Punkte
- die worst-fit-Strategie kann einem mit der kürzesten Antwortzeit einen ausreichend großen Speicherbereich liefern
 - best-fit arbeitet mit konstanter Komplexität und ist deshalb das beste Verfahren
 - first-fit hat konstanten Aufwand bei der Verschmelzung von Lücken
 - bei buddy-Verfahren gibt es keinen externen Verschnitt
- k) Sie kennen den Begriff Demand-Paging. Welche Aussage dazu ist richtig? 2 Punkte
- Demand-Paging setzt eine segmentierte Speicherverwaltung voraus.
 - Demand-Paging benötigt keinerlei Hardware-Unterstützung, da sich alle benötigten Mechanismen auch ohne MMU realisieren lassen.
 - Demand-Paging erlaubt es größere logische Adressräume anzulegen, als Hauptspeicher vorhanden ist. Allerdings muss vorausgesetzt werden, dass ein Prozess nicht alle Seiten des logischen Adressraums tatsächlich anspricht.
 - Demand-Paging lädt eine Seite erst dann in den Hauptspeicher, wenn sie tatsächlich angesprochen wird. Nicht benutzte Seiten werden unter Umständen aus dem Hauptspeicher ausgelagert.

l) Was versteht man unter Virtuellem Speicher? (2 Punkte)

2 Punkte

- Adressierbarer Speicher in dem sich keine Daten speichern lassen, weil er physikalisch nicht vorhanden ist.
- Speicher der einem Prozess durch entsprechende Hardware (MMU) und durch Ein- und Auslagern von Speicherbereichen vorgespiegelt wird, aber möglicherweise größer als der verfügbare physikalische Hauptspeicher ist.
- Einen logischen Adressraum.
- Speicher, der nur im Betriebssystem sichtbar ist, jedoch nicht für einen Anwendungsprozess.

m) Man unterscheidet zwischen privilegierten und nicht-privilegierten Maschinenbefehlen. Welche Aussage ist **richtig**?

2 Punkte

- Privilegierte Maschinenbefehle dürfen in Anwendungsprogrammen grundsätzlich nicht verwendet werden.
- Die Benutzung eines privilegierten Maschinenbefehls in einem Anwendungsprogramm führt zu einer asynchronen Programmunterbrechung.
- Privilegierte Maschinenbefehle können durch Betriebssystemprogramme implementiert werden.
- Mit nicht-privilegierten Befehlen ist der Zugriff auf Geräteregister grundsätzlich nicht möglich.

n) Es gibt verschiedene Ursachen, wie Nebenläufigkeit in einem System entstehen kann (gewollt oder auch ungewollt). Was gehört **nicht** dazu?

2 Punkte

- durch Interrupts
- durch preemptives Scheduling
- durch Traps
- durch Threads auf einem Multiprozessorsystem

Aufgabe 2: (60 Punkte)

Sie dürfen diese Seite zur besseren Übersicht bei der Programmierung heraustrennen!

a) Implementieren Sie ein Programm `lpqmanager`, das Druckaufträge aus einer Menge von Warteschlangen (=Directories) sequenzialisiert und an einen Druck-Server überträgt.

Das `lpqmanager`-Programm erhält als Aufrufparameter eine Liste von Directories, aus denen Aufträge abgearbeitet werden (`lpqmanager dir1 dir2 dir3 ... dirn`).

Das Programm soll folgendermaßen arbeiten:

Queue-Threads:

- Für jedes angegebene Directory gibt es einen Thread (queue-Thread), der in einer Funktion `process_queue` das Directory laufend auf neue Aufträge untersucht. Jeder Auftrag wird durch eine Kontrolldatei repräsentiert, deren Name mit dem Zeichen `c` beginnt. Andere Dateien werden ignoriert. Ist das Directory fertig durchlaufen, wird sofort wieder von vorne begonnen. Sollte beim Öffnen oder Lesen des Directories ein Fehler auftreten, wird eine entsprechende aussagekräftige Fehlermeldung ausgegeben und dieser Thread beendet.
- Die Kontrolldatei wird geöffnet (`fopen`) und der resultierende FILE-Zeiger wird über einen Ringpuffer der Größe `NEBUFFER` (über ein Makro auf 9 voreingestellt) an den Haupt-Thread übergeben. Im Fehlerfall wird der Auftrag einfach ignoriert. Anschließend wird die Kontrolldatei sofort gelöscht (ihr Inhalt bleibt ja, so lange sie geöffnet ist, über den FILE-Zeiger noch lesbar).

Haupt-Thread:

- Der Hauptthread bereitet die Koordinierung auf dem Ringpuffer vor, startet die queue-Threads, baut dann eine TCP-Verbindung zu dem Rechner `printserver.uni-erlangen.de`, Port 515 auf und ruft schliesslich die Funktion `process_jobs` auf.
- Eine Kontrolldatei enthält den vollständigen Pfadnamen einer Datendatei, die die zu druckenden Daten enthält. `process_jobs` entnimmt einen FILE-Zeiger aus dem Ringpuffer und bearbeitet den Auftrag durch Aufruf der Funktion `void job(FILE *datafile)`. Danach werden Kontroll- und Datendatei geschlossen, die Datendatei gelöscht und der nächste FILE-Zeiger aus dem Ringpuffer entnommen.
- Die Funktion `job` sendet zunächst das Kommando `"print job length=N\n"` (N = Größe des Druckauftrags in Bytes) und anschließend den Inhalt der Datendatei über die TCP-Verbindung.

Alle Threads arbeiten endlos und Sie können vereinfachend davon ausgehen, dass die TCP-Verbindung nicht abbricht. Andere Fehler sollen sinnvoll bearbeitet werden (Programmabbruch ist nicht immer sinnvoll, im Zweifelsfall Entscheidung durch Kommentar erläutern!).

Koordinierung:

- Die Koordinierung zwischen den queue-Threads und dem Haupt-Thread erfolgt durch zählende Semaphore. Gehen Sie davon aus, dass Ihnen hierzu ein Modul `sem.o` mit den folgenden Semaphor-Funktionen zur Verfügung steht (Deklarationen in Datei `sem.h`):

```
SEM *sem_init(int n); /* lege neuen Semaphor an, initialisiere
                       ihn mit Wert n und liefere Zeiger
                       darauf zurück, im Fehlerfall NULL */
void P(SEM *s);      /* P und V dekrementieren bzw. */
void V(SEM *s);      /* inkrementieren jeweils um 1 */
```

- Beachten Sie auch, dass zwar nur der Haupt-Thread Aufträge aus dem Ringpuffer entnimmt, aber evtl. mehrere Threads gleichzeitig Aufträge eintragen wollen.

/* Aufträge bearbeiten, Ende main */

.....
.....
.....

/* Ende von main - wird nie erreicht */
return 0;

}
/* Funktion process_jobs */

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

K:
P:

/* Funktion job */

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

.....
.....
.....

P:

Aufgabe 3: (10 Punkte)

Skizzieren Sie in einer programmiersprachenähnlichen Form die Funktionen P und V aus Aufgabe 2. Erläutern Sie die wesentlichen Elemente der Operationen: wo benötigen Sie atomare Operationen bzw. wie oder warum wird an welchen Stellen koordiniert, gewartet, signalisiert, etc.

.....

.....

Aufgabe 4: (20 Punkte)

- a) Skizzieren Sie (graphisch) die Abbildung von einer logischen Adresse in eine physikalische Adresse in einem System mit Segmentierung.
- b) Bei der Erzeugung eines neuen Prozesses müssen die zugehörigen Segmente im Hauptspeicher angelegt werden. Nach welchen Strategien könnte das Betriebssystem die Position bestimmen? Nennen Sie vier solche Strategien und deren Eigenschaften, Vor- und Nachteile.
- c) Wo liegen die wichtigsten Unterschiede zwischen einer segmentierten und einer seitenadressierten Speicherverwaltung?
- d) Zwei Prozesse sollen nun Zugriff auf einen gemeinsamen Hauptspeicherbereich haben. Wie kann das Betriebssystem das in einem segmentierten System realisieren? Was ist hierzu in welchen Datenstrukturen einzutragen?