

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) Welche Art der Kommunikation liegt bei einem write-Aufruf an einem Socket (mit bereits aufgebauter TCP-Verbindung) vor? 2 Punkte
- Synchrone IPC mit sendeseitiger Synchronisation
 - Synchrone IPC mit Client-seitiger Synchronisation
 - Asynchrone IPC mit pufferblockierendem Senden
 - Asynchrone IPC mit unzuverlässigem Senden
- b) In einem Seitendeskriptor werden verschiedene Informationen über eine Seite eines virtuellen Adressraums gehalten. Was gehört sicher **nicht** dazu? 1 Punkt
- Die Benutzererkennung in einem Mehrbenutzersystem
 - Die Position der Seite im Hauptspeicher
 - Zugriffsrechte (z. B. lesen, schreiben, ausführen)
 - Ein Zähler, der ein Maß für das Alter der Seite enthält
- c) Welche Aussage zur Verklemmungsvermeidung (deadlock avoidance) ist richtig? 2 Punkte
- Bei Verklemmungsvermeidung wird dafür gesorgt, dass eine der vier notwendigen Bedingungen für Verklemmungen nicht auftreten kann.
 - Das System überprüft vor dem Freigeben von Betriebsmitteln, ob ein unsicherer Zustand eintreten würde.
 - Mit Hilfe des Bankiers-Algorithmus wird beim Belegen eines Betriebsmittels geprüft, ob mit erfolgreicher Belegung ein unsicherer Zustand eintreten würde.
 - Bei einem Zyklus im erweiterten Betriebsmittelgraph liegt ein unsicherer Zustand vor.

- d) Welcher UNIX-Systemaufruf wird bei der Verwendung von Sockets auf keinen Fall gebraucht? 1 Punkt
- close()
 - accept()
 - mmap()
 - shutdown()
- e) Welche Aussage zum Thema Prozesseinplanung ist richtig? 3 Punkte
- Preemptive Schedulingstrategien bilden die Grundlage zur Implementierung von CPU-Schutz.
 - Kooperative Schedulingstrategien sind vor allem für Mehrbenutzersysteme geeignet.
 - Probabilistische Strategien erlauben eine exakte Vorhersage der CPU-Auslastung
 - Für deterministisches Scheduling ist kein Wissen über das Laufzeitverhalten der beteiligten Prozesse erforderlich.
- f) Gegeben sei folgendes Szenario: zwei Fäden werden auf einem Monoprocessorsystem mit der Strategie "First Come First Served" verwaltet. In jedem Faden wird die Anweisung "i++;" auf die gemeinsame, globale Variable i ausgeführt. Welche der folgenden Aussagen ist richtig? 2 Punkte
- In einem Monoprocessorsystem ohne Verdrängung ist keinerlei Synchronisation erforderlich.
 - Während der Inkrementoperation müssen Interrupts vorübergehend unterbunden werden.
 - Die Inkrementoperation muss mit einer CAS-Anweisung nicht-blockierend synchronisiert werden.
 - Die Operation i++ ist auf einem Monoprocessorsystem immer atomar.

g) Was versteht man unter Virtuellem Speicher?

2 Punkte

- Speicher, in dem sich keine Daten speichern lassen, weil er physikalisch nicht vorhanden ist.
- Speicher, der nur im Betriebssystem sichtbar ist, jedoch nicht für einen Anwendungsprozess.
- 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.
- Unter einem Virtuellen Speicher versteht man einen physikalischen Adressraum, dessen Adressen durch eine MMU vor dem Zugriff auf logische Adressen umgesetzt werden.

h) Beim Blockieren in einem Monitor muss der Monitor freigegeben werden. Warum?

2 Punkte

- weil sonst die Monitordaten inkonsistent sind.
- weil ein anderer Faden die Blockierungsbedingung nur aufheben kann, wenn er den Monitor betreten darf.
- weil kritische Abschnitte immer nur kurz belegt sein dürfen.
- weil der Faden sonst aktiv warten würde.

i) Welche Aussage zu Fäden (Threads) ist **falsch**?

2 Punkte

- Kernfäden können als virtuelle Prozessoren zur Ausführung von Benutzerfäden eingesetzt werden.
- Der Synchronisationsbedarf im Anwendungsprogramm kann von der Ablaufplanung der Kernfäden abhängen.
- Das Betriebssystem kann bei einem durch einen Benutzerfaden ausgelösten Seitenfehler nicht auf einen anderen Benutzerfaden umschalten.
- Das Betriebssystem führt Buch über Kernfäden und Benutzerfäden

j) Wie gross ist die Seitenkacheltable zur Adressabbildung von logischen auf physikalische Adressen in einem System mit Seitenadressierung wenn ein Eintrag in der Seitenkacheltable 32 Bit gross ist, der virtuelle Adressraum 4 GigaByte umfasst und eine Speicherseite 4096 Byte gross ist.

3 Punkte

- 32 Byte
- 512 bis 8.192 Byte
- 1.048.576 Byte
- 4.194.304 Byte

k) 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.

l) Welche Antwort stellt **kein** Attribut einer Datei eines UNIX-UFS-Dateisystems dar?

1 Punkt

- Dateilänge
- Anzahl der symbolischen Links
- Zugriffsrechte
- Eigentümer

m) In welcher Situation tritt kein Seitenfehler auf?

2 Punkte

- Die Seite wurde im Rahmen einer copy-on-write-Datenübertragung einem anderen Prozess zur Verfügung gestellt und der empfangende Prozess greift modifizierend auf die Seite zu.
- Die Seite wurde im Rahmen einer copy-on-write-Datenübertragung einem anderen Prozess zur Verfügung gestellt und der sendende Prozess greift modifizierend auf die Seite zu.
- Die Seite wurde auf Hintergrundspeicher geschrieben, wird in einem Freiseitenpuffer gehalten, ist noch in der Seitentabelle des Prozesses eingetragen und der Prozess greift modifizierend zu.
- Das Referenz-Bit der Seite wurde im Rahmen der Second-Chance-Strategie zurückgesetzt und der Prozess greift danach erneut auf die Seite zu.

n) Bei einer prioritätengesteuerten Prozess-Auswahlstrategie 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.

o) Welche Aussage zum Thema Prozesseinplanung ist **falsch**?

2 Punkte

- Bei Stapelbetrieb ist es ein Hauptziel, die Anzahl fertiggestellter Prozesse pro vorgegebener Zeiteinheit zu maximieren.
- Bei interaktivem Betrieb soll vor allem die Antwortzeit minimiert werden.
- Hohe Systemlast kann bei Echtzeitbetrieb zu einer langsameren Bearbeitung von Prozessen führen.
- Bei allen genannten Betriebsarten wird immer auf eine faire Behandlung aller Prozesse geachtet.

Aufgabe 2: (60 Punkte)

a) Schreiben Sie ein Programm **fileattrd**, das als Server-Prozess Anfragen nach Dateiattributen über TCP/IP beantwortet.

Das **fileattrd**-Programm erhält als Argument die Nummer des Ports auf dem Verbindungen angenommen werden sollen.

Das Programm soll folgendermaßen arbeiten:

- Es nimmt Verbindungen über beliebige IP-Adressen auf dem angegebenen Port an.
- Es wird jeweils eine Verbindung angenommen und in einem Sohnprozess bearbeitet (die Bearbeitung erfolgt in der Funktion `serve(int socket)`), der Vaterprozess steht sofort für die nächste Verbindung wieder bereit.
- Ein Client sendet über die Verbindung die Namen der Dateien (komplette Pfadnamen), deren Attribute er abfragen möchte. Es wird jeweils ein Dateiname in einer Zeile gesendet. Ein Dateiname kann maximal 128 Zeichen lang sein.

- Der **fileattrd**-Server ruft zu jedem angefragten Dateinamen die Funktion `prattr(char *filename)` auf. Die Funktion gibt den Dateinamen, eine Information über den Dateityp (D für Directory, F für "normale Datei" (*regular file*) und X für sonstige Dateitypen) sowie bei normalen Dateien die Dateigröße aus.

Beispiel für eine Ausgabe:

```
/etc/passwd F 2067
/usr/bin D
/dev/tty X
```

- Werden die Attribute einer nicht-existierenden oder nicht-zugreifbaren Datei abgefragt, so wird der Dateiname, gefolgt von der Fehlermeldung ausgegeben.
- Der bearbeitende Sohnprozess terminiert, wenn er alle Daten des Client abgearbeitet hat und der Client die Verbindung geschlossen hat.
- Der Sohnprozess geht beim Terminieren in den Zustand "ZOMBIE" über. Sorgen Sie in Ihrem Programm dafür, dass sich solche terminierten Sohnprozesse nicht ansammeln, sondern dass sie geeignet aufgeräumt werden. Hierfür gibt es verschiedene Möglichkeiten - erläutern Sie die von Ihnen gewählte Lösung durch einen kurzen Kommentar im Programmtext.

Auf den folgenden Seiten finden Sie ein Gerüst für das beschriebene Programm. In den Kommentaren sind nur die wesentlichen Aufgaben der einzelnen, zu ergänzenden Programmteile beschrieben, um Ihnen eine gewisse Leitlinie zu geben. Es ist überall sehr großzügig Platz gelassen, damit Sie auch weitere notwendige Programmanweisungen entsprechend Ihrer Programmierung einfügen können.

Einige wichtige Manual-Seiten liegen bei - es kann aber durchaus sein, dass Sie bei Ihrer Lösung nicht alle diese Funktionen oder ggf. auch weitere Funktionen benötigen.

```

/* includes */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <stdio.h>
#include <errno.h>
#include <signal.h>
#include <unistd.h>
#include <sys/wait.h>

```

```

/* Funktionsdeklarationen, globale Variablen */

```

```

/* Funktion main */

```

```

{
  /* lokale Variablen und was man sonst am
  Anfang so vorbereitet */

```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

```

/* Socket oeffnen */

```

```

/* Socket an angegebenen Port
und beliebige IP-Adressen binden */

```

```

/* Warteschlange ankommender Verbindungen
auf 5 einstellen */

```

S:

