Abschlusstest

(60 Punkte, 90 Minuten)
22.02.2019

_____
Name

_____
Matrikelnummer

☐ Ich stimme zu, dass meine Note mit verkürzter Matrikelnummer auf ISIS veröffentlicht wird.

Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Aufgabe 6	Summe	Note
21	**5**	**12**	**7**	**5**	**10**	**60**	

Einige Hinweise:

- Außer einem Stift sind keine Hilfsmittel zugelassen
- Bitte schreiben Sie alle Lösungen unter Benutzung eines dokumentenechten Stifts auf das Aufgabenblatt
- Wenn Sie weitere Blätter zur Beantwortung der Fragen benötigen, wenden Sie sich bitte an uns
- Wenn Sie die sanitären Örtlichkeiten aufsuchen wollen, wenden Sie sich bitte ebenfalls an uns
- Bitte gehen Sie, wenn in der Aufgabenstellung nicht explizit anders angegeben, stets davon aus, dass die Hard- und Software aus der LV Betriebssystempraktikum betrachtet wird
- Die Klausur besteht aus 11 Seiten

Teil A: Multiple Choice

Klassifizieren Sie die folgenden Aussagen entweder als *wahr* oder als *falsch* durch Ankreuzen!

Teilaufgaben werden als Ganzes gewertet. Es gibt volle Punkte wenn alle Klassifizierungen zutreffen, anderenfalls keine. Nicht klassifizierte Aussagen oder solche, bei denen die Klassifizierung nicht eindeutig erkennbar ist, werden nicht gewertet. Streichen Sie im Zweifelsfall die Kästchen durch und schreiben *wahr* bzw. *falsch* neben die Aussage.

1. Aufgabe (21 Punkte)

1 a) (2 Punkte) Prozessormodi

wahr	falsch	
☐	☐	Der Supervisor-Modus ist ein Ausnahmemodus.
☐	☐	Der User-Modus ist der einzige unprivilegierte Modus.
☐	☐	Der User-Modus kann ausschließlich durch eine Ausnahme verlassen werden.
☐	☐	Der System-Modus kann ausschließlich durch eine Ausnahme verlassen werden.

1 b) (2 Punkte) Modusspezifische Register

wahr	falsch	
☐	☐	Jeder Ausnahmemodus hat ein eigenes Linkregister.
☐	☐	Jeder Ausnahmemodus hat ein eigenes Stackregister.
☐	☐	User- und System-Modus teilen sich dasselbe Stackregister.
☐	☐	Der FIQ-Modus ist der Ausnahmemodus mit den meisten eigenen Registern.

1 c) (2 Punkte) Bei einer Ausnahme ...

wahr	falsch	
☐	☐	... wird das CPSR automatisch im SPSR des entsprechenden Ausnahmemodus gesichert.
☐	☐	... wird das Linkregister des entsprechenden Ausnahmemodus automatisch gesetzt.
☐	☐	... werden einige Register automatisch auf dem Stack des entsprechenden Ausnahmemodus gesichert.

1 d) (2 Punkte) Der Interrupt Controller

wahr	falsch	
☐	☐	... funktioniert nur bei aktivierter MMU.
☐	☐	... steuert die IRQ-Leitung des ARM-Kerns.
☐	☐	... realisiert verschachtelte Interrupts.
☐	☐	... merkt sich bei einem Interrupt den vorherigen Modus und die Rücksprungadresse.

1 e) (2 Punkte) Load/Store-Multiple-Instruktionen (LDM/STM) ...

wahr	falsch	
☐	☐	... können im Supervisor-Modus benutzt werden.
☐	☐	... dürfen nicht im User-Modus benutzt werden.
☐	☐	... erfordern das Abschalten des Caches vor Benutzung.

1 f) (2 Punkte) Der Supervisor-Stack ...

wahr	falsch	
☐	☐	... kann im Supervisor-Modus durch den Befehl `LDR SP, =addr` gesetzt werden.
☐	☐	... muss nach AAPCS stets von kleinen zu großen Adressen wachsen.
☐	☐	... wird beim Betreten einer Ausnahmebehandlung automatisch zurückgesetzt.

1 g) (2 Punkte) Eine L1-Tabelle für die MMU ...

wahr	falsch	
☐	☐	... ist Teil einer übergeordneten Seitentabelle.
☐	☐	... hat genau 4096 Einträge.
☐	☐	... muss der MMU über ihre Adresse bekannt gegeben werden.
☐	☐	... wird von der CPU automatisch mit FAULT-Werten initialisiert.

1 h) (2 Punkte) Der FAULT-Eintrag in einer Seitentabelle ...

wahr	falsch	
☐	☐	... führt zu einem Prefetch/Data Abort beim Zugriff auf die entsprechende Seite.
☐	☐	... ist nur nötig bei aktivem Cache.
☐	☐	... ist nur nötig bei viel RAM (größer 64MB).
☐	☐	... sollte initial für unbenutzte Adressen verwendet werden.

1 i) (2 Punkte) Zur Realisierung von präemptivem Multitasking ...

wahr	falsch	
☐	☐	... ist ein Interrupthandler notwendig.
☐	☐	... muss es möglich sein, mehrere Threads dem selben Adressraum zuzuordnen.
☐	☐	... müssen Threads die CPU regelmäßig mittels Systemruf aufgeben.
☐	☐	... genötigt jeder Thread einen eigenen Stack.

1 j) (3 Punkte) Systemrufe ...

wahr	falsch	
☐	☐	... benötigen rekursive Hardware
☐	☐	... dienen lediglich der Interprozesskommunikation
☐	☐	... erlauben den Zugriff auf privilegierte Ressourcen aus einer unprivilegierten Anwendung heraus.
☐	☐	... funktionieren nur bei Mikrokernelsystemen.
☐	☐	... werden aus dem User-Modus mittels eines Software-Interrupts aufgerufen.
☐	☐	... werden vollständig im User-Modus ausgeführt.

Teil B: Kurzantworten

Für die Beantwortung der Fragen in diesem Teil reichen in der Regel kurze, stichpunktartige Antworten. Sollten Sie aus Platzgründen eine Antwort auf einer anderen Seite fortsetzen, vermerken Sie dies bitte!

2. Aufgabe (Register, 5 Punkte)

2 a) (3 Punkte) Mit welcher Befehlsabfolge kann ein Wechsel von einem privilegierten Modus in einen (beliebigen) anderen privilegierten Modus vorgenommen werden?

(Assemblerbefehle sind nicht Pflicht, jedoch sollte die Antwort jeden in Assembler notwendigen Schritt beschreiben.)

2 b) (2 Punkte) Der folgende Code soll dem *Procedure Call Standard for the ARM Architecture* (AAPCS) folgen. Tragen Sie genau jene Register ein, die auf dem Stack gesichert werden **müssen**. Begründen Sie Ihre Entscheidung!

```
foo:
   stmfd sp!, {_____________________________}
   mov   r7, r0
   add   r1, r7, r7
   mov   r0, r1
   ldmfd sp!, {_____________________________}
   mov   pc, lr
```

3. Aufgabe (Ausnahmen, 12 Punkte)

3 a) (3 Punkte) Mit welcher Befehlsabfolge können Interrupts am ARM-Kern maskiert bzw. demaskiert werden? (Assemblerbefehle sind nicht Pflicht, jedoch sollte die Antwort jeden in Assembler notwendigen Schritt beschreiben.)

3 b) (4 Punkte) Nennen Sie vier Ausnahmen Ihrer Wahl und geben Sie jeweils an, unter welchen Umständen sie durch den ARM-Kern ausgelöst werden!

3 c) (2 Punkte) Erläutern Sie die Notwendigkeit des Software-Interrupts und wofür er im Praktikum eingesetzt wurde!

(Fortsetzung auf nächster Seite)

(Fortsetzung Aufgabe 3)

3 d) (3 Punkte) Benennen Sie die Schritte, die zur erfolgreichen Einrichtung eines Handlers für den Softwareinterrupt durchgeführt werden müssen! Das bezieht sich nicht auf die im Handler notwendigen Schritte, sondern die Schritte, die dazu führen, dass der Handler bei Auftreten eines SWI ausgeführt wird.

4. Aufgabe (MMU, 7 Punkte)

4 a) (3 Punkte) Welche möglichen Zugriffsrechte außer XN können in einem L1-Tabellen-Eintrag für eine im User-Modus laufende Anwendung eingestellt werden? Nennen Sie für jedes Zugriffsrecht außerdem einen sinnvollen Anwendungsfall.

4 b) (2 Punkte) Welche Rolle spielt der TLB bei der Adressumsetzung und wieso muss er gelegentlich invalidiert werden?

4 c) (2 Punkte) Was ist der Unterschied zwischen einem Data- und einem Prefetch-Abort? (2 Punkte)

5. Aufgabe (Geräte, 5 Punkte)

5 a) (1 Punkt) Warum ist bei der Ansteuerung von Geräten in C das `volatile` Keyword notwendig?

5 b) (2 Punkte) Angenommen, es gibt bei einem Gerät auf unserer Zielplattform ein typisches Control-Register CR. Wie kann darin das Bit mit der Nummer nr gesetzt und zu einem späteren Zeitpunkt wieder gelöscht werden? (Lösung als Code-Skizze)

5 c) (2 Punkte) Welche Schritte sind auf unserer Zielplattform notwendig, damit eine Interrupt-Ausnahme ausgelöst wird, wenn das Gerät den entsprechenden Interrupt signalisiert?

Teil C: Komplexe Aufgaben

Für die Aufgaben in diesem Teil gibt es in der Regel mehrere Lösungsmöglichkeiten. Lesen Sie die jeweilige Aufgabe zunächst vollständig und beschreiben Sie ihre Lösung detailliert. Sollten Sie aus Platzgründen eine Antwort auf einer anderen Seite fortsetzen, vermerken Sie dies bitte.

6. Aufgabe (Systemrufe, 10 Punkte)

Jemand hat beschlossen, die Funktionalität von `printf()` in Form eines Systemrufs bereitzustellen. Ob das eine gute Idee ist, sei dahingestellt.

Im Folgenden werden zwei der dabei entstehenden Probleme angegangen:

6 a) (4 Punkte) Definieren Sie eine Aufrufkonvention[1] für Systemrufe, die die Übergabe (nahezu) beliebig vieler Parameter ermöglicht! (Falls notwendig, können Sie annehmen, dass Sie die Anzahl der Parameter kennen.) Achten Sie darauf, dass Ihre Konvention auch noch bei präemptiven Multithreading funktioniert.

(Fortsetzung auf nächster Seite)

[1] „Welchen Parameter finde ich wo?“

(Fortsetzung Aufgabe 6)

6 b) (6 Punkte) Bei der Realisierung des Systemrufs muss der Kernel diverse Male auf Speicheradressen zugreifen, die ihm von einer Anwendung mitgeteilt wurden. Wie kann der Kernel sicherstellen, dass ein solcher Zugriff – gegeben durch Adresse und Datentyp – sicher ist und nicht etwa in einem Data-Abort endet oder das System kompromittiert?