

Betriebssysteme WiSe 2016/17· Übungsblatt 6
Bearbeitungszeit: 19.01.2017 – 02.02.2017 um 09:59 Uhr

Bereiten Sie Ihre **Lösungen** grundsätzlich so vor, dass Sie diese in der Übung Ihren Kommilitonen in geeigneter Form **zeigen** und **diskutieren** können. Geben Sie bitte stets Ihre verwendeten **Quellen** an.

Die Abgabe erfolgt elektronisch über das KVV.

Aufgabe 6-1 (Auswahlstrategien) – 5 Punkte

Um die in der Vorlesung beschriebenen Auswahlstrategien der Speicherzuteilung zu implementieren, muss eine Liste der freien Speicherstücke verwaltet werden. Wie lange dauert die durchschnittliche Suche bei den drei beschriebenen Strategien First-Fit, Next-Fit und Best-Fit?

Aufgabe 6-2 (MMU und Speicherschutz) – 15 Punkte

Nachdem in der vorherigen Aufgabe eine logische Trennung zwischen Anwendungen und Betriebssystem geschaffen wurde, soll in dieser Aufgabe die Memory Management Unit (MMU) in Betrieb genommen werden, um das Betriebssystem auch physisch vor den Anwendungen zu schützen.

Am Ende dieser Aufgabe soll:

1. die MMU eingeschaltet sein,
2. eine L1-Tabelle aufgesetzt sein,
3. der User-Modus nur minimalen Zugriff auf die wirklich benötigten Speicherbereiche verfügen (User-Code lesen, User-Stacks lesen/schreiben),
4. der Kernel nur Zugriff auf belegte Adressbereiche/Speicherbereiche haben,
5. wenigstens ein nicht 1:1-Mapping existieren.

Zur Demonstration soll eine Anwendung verschiedene Zugriffsverletzungen erzeugen:

6. lesender Zugriff auf Null-Pointer,
7. lesender Zugriff auf Kernel-Daten oder -Code,
8. schreibender Zugriff auf eigenen Code,
9. Stack-Overflow,
10. lesender Zugriff auf nicht zugeordnete Adresse.

Außerdem soll demonstriert werden:

11. erfolgreiche lesende/schreibende Zugriffe auf eine Speicherseite an einer logischen Adresse, die ohne MMU zu einem Data-Abort führen würde (Demonstration eines nicht 1:1 Mappings; dies kann z. B. innerhalb der Anwendung mit „sinnlosen“ Zugriffen geschehen, eine andere Möglichkeit wäre ein Verzicht auf den Memory-Remap und das Einblenden einer Speicherseite an Adresse 0 mittels MMU.)

Hinweise:

- Es empfiehlt sich, mit einem reinen 1:1-Mapping mit allen Rechten zu starten. Wenn dies funktioniert, können die Rechte nach und nach eingeschränkt werden und dabei ggf. das Speicherlayout angepasst/überarbeitet werden.
- Einige Lösungsansätze hatten die User-Stacks direkt in den TCBs eingebettet. Dies macht eine Trennung zwischen Anwendung und Kernel schwierig bis unmöglich. Legen Sie den/die Stacks in eigene Speicherseiten und speichern Sie stattdessen im TCB nur einen Zeiger auf den Stack.
- Solange Ihre Anwendung keine globalen Variablen nutzt, muss ich um ein les- und schreibbares User-Datensegment keine Gedanken gemacht werden.
- In dieser Aufgabe geht es nicht um den Schutz zwischen mehreren Anwendungen. Insbesondere ist das Mapping im Moment statisch. Das Ändern des Mappings nach Anschalten der MMU erfordert weitere Maßnahmen, die Teil von Aufgabe 7 sein werden.
- Falls die Anwendung ein `printf()` benötigt, denken Sie daran, dass dies ein anderes `printf()` sein muss als das vom Kernel.