

Portfolioprüfung – Werkstück A – Alternative 2

Aufgabe 1 Aufgabe

Entwickeln und implementieren Sie einen Simulator zur Visualisierung verschiedener Realisierungskonzepte für das Speicherverwaltungskonzept **dynamische Partitionierung**.

Der Simulator soll auf Kommandozeilenebene die Realisierungskonzepte **First Fit**, **Best Fit**, **Next Fit**, **Worst Fit** (das Gegenteil von Best Fit) und **Random** für dynamische Partitionierung visualisieren.

Entwickeln und implementieren Sie Ihre Lösung als Bash-Skript als freie Software (Open Source) und verwenden Sie hierfür ein Code-Repository, z.B. bei GitHub.

Bearbeiten Sie die Aufgabe in Teams zu **4 Personen**.

Schreiben Sie eine aussagekräftige und ansehnliche Dokumentation (Umfang: **8-10 Seiten**) über Ihre Lösung.

Bereiten Sie einen Vortrag mit Präsentationsfolien und eine Live-Demonstration (Umfang: **15-20 Minuten**) vor. Demonstrieren Sie die Funktionalität der Lösung in der Übung.

Aufgabe 2 Anforderungen an den Simulator

- Das fertige Programm soll eine Kommandozeilenanwendung sein.
- Der Quellcode soll durch Kommentare verständlich sein.
- Benutzer sollen die Größe des gesamten Speichers über eine Benutzereingabe (interaktiv) oder per Kommandozeilenargument definieren, also z.B. `-size 1024`.
- Benutzer sollen das Realisierungskonzept First Fit, Best Fit, Next Fit, Worst Fit (das Gegenteil von Best Fit) oder Random über eine Benutzereingabe (interaktiv) oder per Kommandozeilenargument definieren, also z.B. `-concept <name>`.
- Benutzer sollen über eine Benutzereingabe (interaktiv) über Tastatureingaben (siehe Abschnitt 3) nach Belieben das Starten und Beenden von Prozessen simulieren können.
- Ein erstellter (simulierter) Prozess soll je nach Realisierungskonzept eine freie Partition belegen. Wenn das nicht möglich ist, weil keine ausreichend große

freie Partition existiert, muss der Simulator entsprechend mit einer Fehlermeldung reagieren.

- Die Belegung des Speichers soll auf der Kommandozeile jederzeit (zumindest nach jeder Benutzerinteraktion) visualisiert werden. Es soll jederzeit erkennbar sein, welche Speicherbereiche im Adressraum belegt und welche frei sind.
- Nach jeder Änderung der Speicherbelegung soll der Simulator Informationen zur Speicherbelegung in der Kommandozeile ausgeben. Sinnvolle Informationen sind u.a.:
 - Grad der externen Fragmentierung
 - Größter/kleinsten freier Speicherblock
 - Gesamtzahl belegter/freier Blöcke im Adressraum

Aufgabe 3 Steuerung des Simulators

- Die Steuerung des Simulators soll nicht über einzutippende Befehle geschehen! Die Benutzer sollen mit den Pfeiltasten in der Lage sein zwischen den Befehlen
 - **create** (Partition anlegen),
 - **delete** (Partition löschen),
 - **new** (Speicherbelegung zurücksetzen) und
 - **quit** (Simulator beenden)

auszuwählen. Das Bestätigen eines Befehls geschieht mit der Return-Taste.

- Alternativ können Sie auch Shortcuts definieren (z.B. Die Tasten **c** und **d**), um die Befehle **create** und **delete** auszuwählen.
- Hat ein Benutzer den Befehl **create** ausgewählt, wird er nach dem Namen des neuen Prozesses und nach der Größe gefragt. Diese Namen bzw. Werte kann der Benutzer mit der Tastatur eingeben und mit dem Befehl **create** bestätigen.
- Hat ein Benutzer den Befehl **delete** ausgewählt, wird er nach dem Namen des zu löschenden Prozesses gefragt. Diesen Namen kann der Benutzer mit der Tastatur eingeben und mit dem Befehl **create** bestätigen.
- Nach jedem erfolgreichen Befehl werden im Anschluss die Belegung des Speichers visuell und die Informationen zur Speicherbelegung ausgegeben.
- Fehlerhafte Eingaben werden abgefangen und führen zu einer entsprechenden Fehlermeldung sowie zur Ausgabe der Belegung des Speichers visuell und der Informationen zur Speicherbelegung.

Aufgabe 4 Literatur

- Foliensatz 2 der Vorlesung **Betriebssysteme und Rechnernetze** im SS2022
- **Betriebssysteme kompakt**, *Christian Baun*, 2. Auflage, Springer Vieweg (2020), S. 79-82
- **Operating Systems / Betriebssysteme: Bilingual Edition**, *Christian Baun*, 1. Auflage, Springer Vieweg (2020), S. 77-80
- **Betriebssysteme – Prinzipien und Umsetzung**, *William Stallings*, 4. Auflage, Pearson (2003), S. 362-365
- **Operating Systems – Internals and Design Principles**, *William Stallings*, 4. Auflage, Prentice Hall (2001), S. 309-313