3te SMÜ – Fragen

1. **Speicherhierarchie, Zugriffzeiten X**

Register: ns

Cache: ns

Hauptspeicher: ns

Platten, SSD, NVMe, PMEM: ms

Bandlaufwerk, DVD: ms bis s

1. **Lokalität: zeitlich, räumlich, Bspe X**

**Zeitlich:** Im zeitlichen Ablauf wird häufig auf dieselben Speicheradressen zugegriffen, darum werden Speicheradressen bei ihrem Zugriff in den Cache ausgelagert

**Räumlich:** nach Zugriff auf Speicheradresse, wird nächster Zugriff auf eine Adresse in unmittelbarer Nähe folgen, z.B. bei Workset eines Prozesses

1. **Adressraumbelegung durch eine App X**

Festlegung durch Compiler und Laufzeitumgebung

1. **Speicherbelegung einer App: zwei unterschiedliche Modelle**

* Programmcode, Konstante, intialisierte Daten, nicht initialisierte Daten; Heap -> Guard <- Stack
* Guard <- Heap, Programmcode, Konstante, intialisierte Daten, nicht initialisierte Daten; Stack -> Guard

1. **Overlay Technik, warum, wie X**

**Warum:** Um RAM zu sparen

**Wie:** Programme werden durch Programmierer in Teile zerlegt, die nicht alle gleichzeitig im RAM sind. Falls ein Teil des Programmes gebraucht wird der sich gerade nicht im RAM befindet wird dieser nachgeladen und durch einen anderen Teil wiederrum ersetzt.

1. **Warum Speicherschutz? X**

Damit ein Prozess nicht in den Speicherbereich eines anderen schreiben oder lesen kann.

1. **Speicherverwaltung mit festen Partitionen. Erkläre die Funktionsweise X**

Arbeitsspeicher wird in feste Teile mit einer fixen Menge an Speicher in sogenannte Partitionen eingeteilt. In jeder Partition wird dabei genau ein Programm ausgeführt.

1. **Speicherverwaltung mit variablen Partitionen, Memory Compaction. Erkläre die Funktionsweise X**

Prozesse bekommen Speicher den sie benötigen. Um einen neuen Prozess ausführen zu können, muss ein anderer ausgelagert werden.

1. **Müssen alle Daten eines Prozesses im RAM sein, oder wo können sie noch sein? X**

Nein, sie können sich auch in einem Page File eines Sekundärspeichers befinden.

1. **Grundprinzip der virtuellen Speichertechnik. Was ist eine Page, ein Page Frame und wie groß sind diese – sind sie gleich X**

Ein **Page** ist ein virtueller Block eines Prozesses

Der **Page Frames** sind Frames, Kachel oder Seitenrahmen mit der Größe 1, 4, 8, 16, 64 KiB

und beide sind gleich groß

1. **Strategien für das RAM-Management. Mit Erklärung**

* **Paging / demand Paging:** Pages werden in PageFile.sys ausgelagert, nur benötigte Seiten werden in den RAM eingelagert.
* **Austauschstrategie:** Welche Teile werden beim Verdrängen entfernt?
* **Aufräumstrategie:** Sorgt dafür, dass immer wieder Platz im realen Speicher geschaffen wird.
* **Abrufstrategie:** Regelt, wann Speicherbereiche in den realen Speicher eingelagert werden.

1. **Was ist die MMU und welche Aufgabe hat sie? X**

MMU = Memory Management Unit, ihre Aufgabe ist es, physikalische Adressen in virtuelle Adressen umzuwandeln und umgekehrt.

1. **Einfach Adressumsetzung**

MMU berechnet aus dem Seitentabellenindex und der Distanz (Offset) der virtuellen Adresse, die reale Adresse, welche aus der Framenummer und dem Umgerechneten Offset besteht.

1. **Wie erfolgt die Adressumsetzung**

indem die MMU virtuelle Adressen in reale umwandeln muss, die virtuelle Page muss dabei im RAM sein, sonst kommt es zu einem Page Fault(Trap)

1. **Zweistufige Seitentabelle, Erklärung**

Jede Seitentabelle verweist auf eine weitere Seitentabelle, außer die letzte, welche auf den Frame verweist.

1. **Seitentabelleneintrag – detailliert**

* Caching C: on/off
* Reference-Bit R: Gibt an, ob ein Zugriff erfolgt.
* Modify-Bit M: Gibt an, ob eine Seite verändert wurde (Dirty Bit)
* Protection: Zugriffsschutz(lesen, schreiben, ausführen)
* Present/Absent-Bit P/A: ist die aktuelle Seite im Frame?
* Verweis Frame-Nummer f: auf Frame im RAM

1. **Was ist ein shared Memory**

Speicherbereich, der nur einmal geladen wird und von mehreren Prozessen genutzt werden kann.

1. **Was ist SLAT?**

SLAT = Second Level Address Translation

Ist eine hardwareunterstützte Virtualisierungstechnologie die eine effizientere Umsetzung von Shadow Page Tables ermöglicht.

1. **Warum konnte man bei einem 32 Bit Prozessor nur 3 GB RAM adressieren. Wie groß war der Adressbuss**

* Man konnte nur 3 GB adressieren weil 1 GB für DMA-Geräte reserviert war.
* 32 Bit