

1. Fragen zu den Prozess-Grundlagen

- a) Erklären Sie, weshalb Prozesse ein zentrales Betriebssystem-Konzept darstellen

Lösungsvorschlag:

Prozesse bilden die Grundlage zur Ausführung von Programmen jeglicher Art. Ein Prozess kann dabei als die laufende Instanz eines Benutzerprogrammes betrachtet werden. Erst durch die Verwendung von Prozessen können Multiprogrammiersysteme und damit Quasi-Parallelität bei Einkern-Prozessorsystemen umgesetzt werden.

- b) Erläutern Sie den Unterschied zwischen CPU-Rechenzeit und I/O-Wartezeit hinsichtlich eines Prozesses

Lösungsvorschlag:

Erhält ein Prozess in einem einfachen Multiprogrammiersystem die CPU zugeteilt, so steht ihm diese für einen fest definierten Zeitslot zur Verfügung. Man bezeichnet den Zeitanteil, in dem der Prozess innerhalb des Zeitslots Berechnungen durchführt, als CPU-Rechenzeit.

Den Rest der Zeit verbringt der Prozess mit z. B. dem Warten auf eine Tastatureingabe, oder auf Daten von der Festplatte.

Diesen Anteil bezeichnet man als I/O-Wartezeit.

Der Prozess „blockiert“ währenddessen die CPU für andere rechenwillige Prozesse, obwohl er keine Berechnungen durchführt → ineffiziente CPU-Nutzung.

- c) Wie kann ein Betriebssystem die Prozessor-Ausnutzung effizient gestalten, sodass die Gesamt-CPU-Rechenzeit erhöht wird?

Lösungsvorschlag:

Indem wartenden / blockierenden Prozessen die CPU entzogen wird und dem nächsten rechenbereiten Prozess die CPU zugeteilt wird. Dadurch erhalten ausschließlich rechenbereite Prozesse die CPU zugeteilt, sodass Leerlaufzeiten auf ein Minimum reduziert werden

1. Fragen zu den Prozess-Grundlagen

- d) Stellen Sie die Begriffe Prozess und Programm anhand von drei Merkmalen einander gegenüber

Lösungsvorschlag:

Ein Prozess stellt eine Aktivität dar, das Programm ist das „Rezept“ / die Ausführungsbeschreibung für diese Aktivität.

Ein Programm / eine Applikation wird i. d. R. auf dem Festspeicher abgelegt. Der Prozess hingegen ist flüchtig, d. h. seine benötigten Ressourcen werden im Hauptspeicher abgelegt, bis sie auf den Festspeicher niedergeschrieben werden.

Beziehung (Programm, Prozess) = 1 : (1 ... *)

- e) Nennen Sie drei Prozess-Ressourcen

Lösungsvorschlag:

Register, Befehlszähler, Stackpointer, offene Fehlersignale, verbundene Prozesse, ...

2. Fragen zur Prozesserzeugung und –beendigung

- a) Nennen Sie zwei verschiedene Möglichkeiten zur Prozesserzeugung und erklären Sie diese mit jeweils einem Beispiel

Lösungsvorschlag:

Bei System-Initialisierung: z. B. ein Systemtreiber für ein Peripheriegerät

Systemaufruf durch anderen Prozess:

Z. B. wenn ein Tabellenverarbeitungsprogramm aus einer Benutzeranwendung heraus gestartet werden soll

Mittels Benutzeranfrage zur Prozesserzeugung: z. B. Klick auf ein Anwendungssymbol

- b) Nennen Sie zwei verschiedene Möglichkeiten zur Prozessbeendigung und erklären Sie diese mit jeweils einem Beispiel

Lösungsvorschlag:

Normales Beenden: z. B. ein Programm, das nach Gebrauch durch den Benutzer beendet wird

Freiwilliges Beenden aufgrund eines Fehlers: z. B. wenn ein Timeout für eine Verarbeitungsroutine erreicht wurde und Ressourcen für andere Applikationen freigegeben werden müssen

Unfreiwilliges Beenden aufgrund eines Fehlers: z. B. wenn das Betriebssystem die weitere Programmausführung beendet, weil eine Speicherzugriffsverletzung erfolgt ist

2. Prozesserzeugung und –beendigung

- c) Erläutern Sie, inwiefern sich die Adressraumverwaltung auf einem UNIX-Betriebssystem von der eines Windows-Betriebssystems nach der Prozesserzeugung unterscheidet (Stichwort: Copy-on-Write Prinzip)

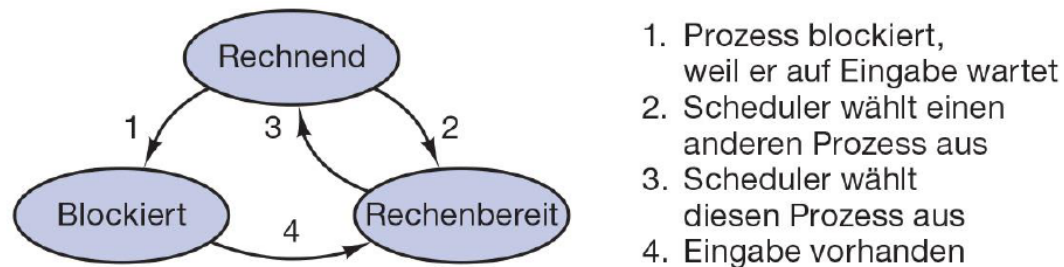
Lösungsvorschlag:

Das Copy-on-Write Prinzip ist eine Möglichkeit zur Separierung bestimmter Speicherbereiche, wenn sich Eltern- und Kind-Prozesse denselben Adressraum teilen. Bei der Veränderung eines Speicherblocks durch einen der Prozesse wird dabei der betreffende Speicherblock in einen privaten, dem Prozess zugeordneten Bereich kopiert. Anschließend werden die gewünschten Änderungen in diesem isolierten Speicherbereich vorgenommen, der nur für den entsprechenden Prozess einsehbar ist. Der andere Prozess hingegen sieht nach wie vor die Ursprungsdaten. Auf diese Art und Weise erfolgt die Adressraumverwaltung unter UNIX-basierten Betriebssystemen. Unter Windows-Systemen hingegen wird bei der Erzeugung eines Kind-Prozesses der komplette Adressraum dupliziert und separiert vom Eltern-Prozess für den Kind-Prozess bereitgestellt.

3. Prozesszustände

- a) Zeichnen Sie einen Graphen, in dem Sie alle möglichen Prozesszustände als Knoten und mögliche Prozessübergänge als Kanten darstellen und beschriften Sie sowohl Knoten als auch Kanten aussagekräftig

Lösungsvorschlag:



- b) In welchem Zustand befindet sich ein Prozess, wenn er nicht rechenbereit ist, weil er z. B. auf ankommende Netzwerkdaten oder eine Tastatureingabe wartet?

Lösungsvorschlag:

Der Prozess befindet sich in diesem Fall im Zustand „Blockiert“.