BS // Fragebogen

Programmfäden

1. Was ist eine Koroutine und welcher Unterschied besteht zu einer Routine (bzw. Prozedur, Funktion oder einem Unterprogramm)? Welche Gemeinsamkeiten und Unterschiede bestehen zwischen den Konzepten?

2. Was ist ein Aktivierungsblock (*activation record*) und wozu dient er? Welche Ebene (d.h. virtueller und/oder realer Maschine) legt Aufbau und Struktur des Aktivierungsblocks fest?

3. Woraus setzt sich der Laufzeitkontext eines Programmfadens zusammen?

4. Welche fundamentalen Eigenschaften besitzt die `resume`-ELOP? Wie erfolgt der Kontextwechsel technisch und wann gilt er als vollzogen?

5. Was verbirgt sich hinter der Kontextvariablen einer Koroutine? Welche artspezifischen Unterschiede ergeben sich für den Kontextwechsel?

6. Welcher Ebene (d.h. virtueller und/oder realer Maschine) wird die `resume`-ELOP typischerweise zugerechnet? Welche anderen Ebenen kämen noch in Frage?

7. Weshalb ist die Instanzenbildung von Koroutinen erforderlich und welche Aufgaben nimmt sie war? Warum ist keine spezielle ELOP zur Zerstörung einer Koroutine notwendig?

8. Weshalb ist eine „Notbremse“ vorzusehen, um eine normale Prozedur technisch als Koroutine auffassen zu können? Welche Aufgabe(n) muss die dafür notwendige Software typischerweise erfüllen?

9. Warum ist die Berechung des initialen Wertes für den Stapelzeiger einer zu instanziierenden Koroutine maschinenabhängig?

10. Inwiefern unterscheiden sich „nebenläufige Blöcke“ und „Prozessabzweigung“ vom Koroutinenkonzept?

11. Welches grundsätzliches Problem ergibt sich mit den Fadenmodellen herkömmlicher Programmiersprachen, wenn diese ggf. genutzt werden sollen, um die Prozessverwaltung eines Betriebssystems zu implementieren? Was darf durch den Programmierspracheneinsatz keinesfalls geschehen?