

# Übungsblatt 1

## Betriebssysteme im WiSe 2020/2021

### Zum Modul A

- Abgabetermin:** 08.11.2020, 18:59 Uhr
- Besprechung:** Besprechung der Übungsaufgaben in den Übungsgruppen vom 16. – 20. November 2020
- Ankündigungen:** Herzlich willkommen zum Übungsbetrieb zur Vorlesung Betriebssysteme. Bitte melden Sie sich zu den Übungsgruppen im *Uni2Work* **bis zum 8. November 2020 18:59 Uhr** an. Eine spätere Anmeldung ist nicht mehr möglich. Beachten Sie dazu auch die Hinweise auf dem Merkblatt. Um kurzfristige Ankündigungen nicht zu verpassen, bitten wir Sie auch regelmäßig die *Website zur Vorlesung* zu besuchen.

### Aufgabe Ü1: Unterprogramme

(8 Pkt.)

- Nennen Sie zwei Nachteile, die sich ergeben, wenn ein Programmierer ausschließlich offene Unterprogramme verwendet.
- Offensichtlich ist es sehr ineffizient, auf geschlossene Unterprogramme (Prozeduren) ganz zu verzichten. Bei besonders kleinen Unterprogrammen ist es aber wiederum ungünstig, diese als geschlossene Unterprogramme zu implementieren. Warum?
- Welche grundsätzlichen Arten der Parameterübergabe gibt es?
- Wie werden Sprünge innerhalb eines Programms technisch realisiert?
- Was ist der wesentliche Unterschied zwischen Sprüngen, die mit den Befehlen `JMP` und `CALL` eingeleitet werden?
- Welche zwei wesentlichen Möglichkeiten gibt es, den `RET`-Befehl zu implementieren und woraus ergeben sie sich?

### Aufgabe Ü2: Rekursive Prozeduraufrufe

(6 Pkt.)

Die Elemente der Fibonacci-Folge

0, 1, 1, 2, 3, 5, 8, 13, ...

können durch eine rekursive Bildungsvorschrift berechnet werden:

$$F_n = F_{n-1} + F_{n-2} \text{ für } n > 1, \text{ mit den Anfangswerten } F_0 = 0, F_1 = 1$$

Dafür sei folgende Prozedur gegeben:

```

PROCEDURE Fib(n: INTEGER) RETURNS INTEGER
BEGIN
  IF n <= 1 THEN RETURN n
  ELSE RETURN Fib(n-1) + Fib(n-2)
END

```

Gehen Sie davon aus, dass aus dem Hauptprogramm einen Aufruf `Fib(4)` erfolgt.

Erstellen Sie in Anlehnung an das Beispiel 2.2 auf Seite 25 des Skriptes ein Diagramm, aus welchem ausgehend vom Hauptprogramm ersichtlich wird, wann welcher Unterprogrammaufruf erfolgt, welcher Parameterwert übergeben wird, wie der Rückgabewert aus dem Unterprogrammaufruf lautet und was das Endergebnis der Berechnung ist. Sie können dabei davon ausgehen, dass Ausdrücke in der Prozedur von links nach rechts ausgewertet werden.

### Aufgabe Ü3: Einfachauswahlaufgabe: Das Betriebssystem

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Nennen Sie dazu in Ihrer Abgabe explizit die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

a) Welche logische Hierarchieebene liegt direkt unter der Ebene des Betriebssystems?			
(i) Physikalische Geräte	(ii) Mikroprogrammierung	(iii) Anwendungsprogramme	(iv) Maschinsprache
b) Was ist keine direkte Aufgabe des Betriebssystems?			
(i) Textverarbeitung	(ii) Komplexitätsreduktion	(iii) Ressourcenverwaltung	(iv) Zugriffskontrolle
c) Was ist kein Betriebssystemtyp?			
(i) Mainframe-Betriebssysteme	(ii) PC-Betriebssysteme	(iii) Echtzeit-Betriebssysteme	(iv) Analog-Betriebssysteme
d) Was stellt allgemein eine Schnittstelle zwischen Anwendungsprogrammen und Betriebssystem dar, durch die ein Anwendungsprogramm auf eine Ressource zugreifen kann, auf die es keinen direkten Zugriff hat?			
(i) Shared Memory	(ii) Systemaufrufe	(iii) Sockets	(iv) Pipes
e) Welche der folgenden Antworten zeigt die korrekte Implementierung des RET-Befehls, wenn die Rückkehradresse auf dem Stack gespeichert wurde und PC für Program Counter steht?			
(i) COMMAND RET BEGIN PC := POP; END	(ii) COMMAND RET BEGIN PC := PUSH; END	(iii) COMMAND RET BEGIN PUSH (PC); END	(iv) COMMAND RET BEGIN PC := PC + 1; END