Übung 2

Mittwoch, 6. Mai 2020 15:58

3. Theoretische Fragen

3.1 Konstruktoren und Destruktoren

main1;
Anton ctor;
Berta ctor;
main2;
Anton body;
main3;
Berta dtor;
Anton dtor;

3.2 Gültigkeitsbereiche von Variablen

- das Objekt Anton wird an Stelle 15 auf dem Stack gespeichert
- damit zeigt front auf die Stelle 15 vom Stack
- Berta wird an der gleichen Stelle gespeichert
  -> Adresse, auf die front zeigt, wird überschrieben
  -> Aufruf quasi von berta.body()

3.3 Kontrollfragen

1. Was ist ein Prozess im Sinne der Informatik? Durch welche Teile definiert er sich?
   - Programm in Ausführung
   - eigener Laufzeitkontext
   - PROZESS DES KOCHENS
   - bezeichnet ein im Ablauf befindliches Computerprogramm
   - besteht aus
     - Programm samt Daten
     - Prozesskontext
   - Prozess P ist ein Tripel (S, f, s)
     - S = Zustandsraum
     - f = Aktionsfunktion
     - s ⊂ S = Anfangszustände des Prozesses P

2. Warum ist die Abstraktion eines Prozesses sinnvoll? Betrachte hierbei den Aspekt der Monopolisierung einer CPU!
   -

3. Was versteht man konkret unter einem Prozesswechsel (welche notwendigen Schritte müssen unternommen werden)?
   - Wechsel von einem Prozess zu einen anderen
   - schwergewichtete Prozesse
     - erfordern Adressraumwechsel, da eigener Adressraum
   - leichtgewichtige Prozesse(threads)
     - kein Adressraumwechsel nötig
   - Adressraum
     - besteht aus 4 logischen Speicherbereichen
     - Code- oder Textsegment
     - Datensegment
     - Heapsegment
     - Stack

4. Wie sieht ein Prozesswechsel dementsprechend auf der x86 Hardware aus?

```
switchContext:
;»  fuegt hier Euren Code ein!
»   push ebp
»   mov ebp, esp
»   »
»   push edi
»   push esi
»   push ebx
»
»   mov eax, [ebp + 8]
»   mov [eax], esp
»   mov eax, [ebp + 12]
»   mov esp, [eax]
»
»   pop ebx
»   pop esi
»   pop edi
»   pop ebp

;Ende
»   ret»»   ; Ruecksprung zum Aufrufer
```

5. Wie sieht der Stack beim ersten Wechsel zu einem Prozess aus?
   - das ist der Stack von der main

6. Was ist ein Prozesskontrollblock und was beschreibt er?
   - enthält alle zu einem einzelnen Prozess gehörenden Verwaltungsinformationen
     - Prozess-ID
     - CPU-Register
     - Ausführungszustand
     - Adressraum
   - sobald neuer Prozess: neuer PCB wird als Verwaltungsstruktur angelegt
   - für jeden Prozess existiert somit ein eigener PCB

7. In welchem Kontext macht ein Prozesskontrollblock Sinn?
   - wenn man viele Prozesse hat und diese verwalten muss
   - wenn man die Reihenfolge der Prozessaufrufe steuern will

8. Welche Arten von Prozessverwaltung gibt es?
   - User-Level Threads
     - Verwaltung im User-Space
     - BS kennt nur den Prozess, nicht die Threads
   - Kernel-Level Threads
     - BS verwaltet Threads

9. Nenne und erkläre grundlegende Algorithmen der Prozessverwaltung!
   - Dispatcher
   - Worker

10. Was ist eine Ready-Liste und wozu dient sie?
    - beinhaltet alle lauffähigen Prozesse
    - wenn ein Prozess den Löffel abgibt, weiß die Liste, welcher Prozess als nächstes kommt

11. Müssen Prozesse beendet werden? Wenn ja, wann und wie? Wenn nein, warum nicht? Wie lange existieren Prozesse dann?
    - ja
      - selbstständig, erklärt sich mittels Systemaufruf als beendet
      - unselbstständig
        - aus Warteschlange gelöscht
        - von anderem Prozess beendet
    - sollte vor Ende sämtliche Dateien schließen und jegliche Ressourcen zurückgeben

12. Was ist eine Coroutine?
    - Basis aller Prozesse
    - "Prozeduren" mit eigenem Laufzeitkontext
      - Kontrollfluss kann von einer Coroutine zur anderen explizit transferiert werden
    - viel mächtiger als Prozeduraufrufe
      - Kontrolltransfer nicht hierarchisch eingeschränkt wie bei Prozeduraufrufen
      - unabhängige Aktivitäten statt Prozeduren
      - Aktivitäten können explizit suspendiert und wiederaufgenommen werden
    - transferieren den Kontrollfluss