

1. Das Betriebssystem kann als 'Virtuelle Maschine' sowie als 'Ressource Manager' betrachtet werden. Erklären Sie beide Begriffe. Was gilt zu beachten?
 - 'Virtual machine' weil es Hardware Komponenten nimmt, und in einem generellen, einfach zu verwendeten Teil umwandelt
 - 'Resource manager' weil es ermöglicht mehreren Programmen Ressourcen zu teilen
2. In welchem Modus wird der OS Kernel ausgeführt. Welche weitere Modi kennen Sie ?
 - OS Kernel wird in 'kernel mode' ausgeführt
 - Es gibt noch 'user mode' wo die Benutzeranwendungen laufen
3. Was ist ein Systemcall? Wie unterscheidet sich ein Systemcall von einem Funktionscall
 - Ein Systemcall ist von Programme benutzte Methode um vom Betriebssystem bereitgestellten Funktionalitäten auszuführen
 - Der Unterschied liegt darin, dass ein Systemcall die Kontrolle überträgt und dabei die Berechtigungsstufe ändert. Von user mode zu kernel Mode zum Beispiel.
4. Was ist der Unterschied zwischen Policy und Mechanismus?
 - Policy ist ein Entscheidungsträger um eine Aufgabe zu optimieren
 - Mechanism ist low level Code, dass die Entscheidung implementiert
5. Was ist ein Prozess? Was gehört alles zum 'Process Machine State' ?
 - Adressbereich, also Speicher das ein Prozess adressieren kann (heap, stack und code)
 - Alles was von laufenden Code beeinflussen wird und werden kann
 - Geöffnete Dateien
6. Wie wird ein Prozess gestartet?
 - OS lädt Code und statische Daten ins Speicher und Adressbereich
 - Während Ausführung werden wenn gebraucht Code und Daten geladen
 - Speicher für das Prozess allokiert und dann 'file descriptors' initialisieren (I/O)
 - OS springt dann zu der main() Routine
7. Was ist eine 'Direct Execution'? Warum wird diese nicht angewendet?
 - Erlaubt Benutzer ein Prozess direkt auf den Hardware auszuführen
 - Probleme sind:
 - Prozess könnte was beschränktes machen (Daten lesen/ schreiben)
 - Prozess könnte für immer laufen (langsam, buggy, böartig). OS möchte die

Kontrolle haben zwischen Prozesse zu wechseln

→ Prozess könnte ineffizient sein. OS möchte Ressourcen (CPU) effizient nutzen

8. Welche 3 Basis Prozesszustände gibt es? Welche Ereignisse führen zu einer Transition des Prozesszustandes?

- Running
- Ready
- Blocked

9. Zeigen Sie im 'process-run.py' Simulator, wie sich der Schalter -S auf CPU und I/O Prozesse auswirkt.

- Einstellungen SWITCH_ON_IO, SWITCH_ON_END
- Also wenn IO startet, sollte man auf anderen Prozess wechseln, oder warten bis das IO fertig wird

10. Betrachten Sie den folgenden Code:

```
int *p; //a1
p = malloc(sizeof(int)); //a2
assert(p != NULL);

printf("(pid:%d) addr of p:      %llx\n", (int)
        getpid(), (unsigned long long) &p);
printf("(pid:%d) addr stored in p: %llx\n", (int)
        getpid(), (unsigned long long) p);
*p = atoi(argv[1]); //a3
while (1) {
    Spin(1);
    *p = *p + 1;
    printf("(pid:%d) value of p: %d\n",
            getpid(), *p);
}
```

- In welchem Speicherbereich liegen die Variablen mit den Kommentaren a1 und a2
 - a1 liegt auf dem Stack
 - a2 auf dem Heap