Was ist ein Prozess im Sinne der Informatik? Durch welche Teile definiert er sich?

Process is a program under execution. It include the program executed, associated data and stack space Processor is the hardware that executes the program. It is the CPU of computer.

Warum ist die Abstraktion eines Prozesses sinnvoll? Betrachte hierbei den Aspekt der Monopolisierung einer CPU!
https://www4.cs.fau.de/Lehre/WS02/V_BS/folien/10-A6.pdf

Um nicht jede Prozess an sich behandeln sonder generell Type von Prozessen zu diskutieren und abstrahieren von den feinen Details in die generellen Idee. Da Im prozessor- Milliarden von Prozessen gibt sollten wir generell denken und nicht an jedes Programm denken. Wir können eine richtige Zuteilung der Privileg haben wenn wir typisieren können welchen Prozesse mit Sonder Rechts ausgeführt werden

Monopolisierung einer CPU heißt dass das Betriebssystem die Kontrolle verliert und die CPU nur Anweisung führt, die vom Benutzerprogramm vorgibt.
Um das zu fixen sollte

1. die Anweisungen asynchron geführt. d.h. mit Interrupts
2. Zugriffe auf Zeitgeber und Interrupt-Maske sind privilegierte Operationen

Was versteht man konkret unter einem Prozesswechsel (welche notwendigen Schritte müssen unternommen werden)?
https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.050/vorlesungen/sose14/gdbs/GdBS-falk-SS14-4_Prozesse.pdf
Switching the CPU to another process requires **saving** the state of the old process and **loading** the saved state for the new process. This task is known as a **Context Switch**.

Sichern der Register des laufenden Prozesses, inkl. Programmzähler (Kontext)
– Auswahl des neuen Prozesses
– Ablaufumgebung des neuen Prozesses herstellen (z.B. Speicherabbildung, etc.)
– Gesicherte Register laden, und
– Prozessor aufsetzen, d.h. Programmzähler laden

Wie sieht ein Prozesswechsel dementsprechend auf der x86 Hardware aus?
The x86 CPU provides a way of doing it completely in hardware,
you need to tell the CPU where to save the existing CPU state, and where to load the new CPU state.
To trigger a context switch and tell the CPU where to load it's new state from the far version of CALL and JMP instructions are used.
The CPU has a register called the "TR" (or Task Register) which tells which TSS will receive the old CPU state. When the TR register is loaded with an "LDTR" instruction the CPU looks at the GDT entry (specified with LDTR) and loads the visible part of TR with the GDT entry, and the hidden part with the base and limit of the GDT entry. When the CPU state is saved the hidden part of TR is used.

Wie sieht der Stack beim ersten Wechsel zu einem Prozesses aus?

Was ist ein Prozesskontrollblock und was beschreibt er?
http://www.inf.fu-berlin.de/lehre/WS11/OS/slides/OS_V5_Prozessverwaltung_.pdf
Für die Prozesssteuerung verwendet das Betriebssystem eine Prozesstabelle. Dort befindet sich ein Eintrag pro Prozess. Dieser Eintrag wird oft Prozesskontrollblock genannt

https://de.wikipedia.org/wiki/Prozesskontext
is a data structure used by computer operating systems to store all the information about a **process**. When a process is created (initialized or installed), the operating system creates a corresponding process control block.

In welchem Kontext macht ein Prozesskontrollblock Sinn?
Der Prozesskontrollblock stellt die wichtigste Datenstruktur des Betriebssystems dar.
it includes the value of the CPU registers, the process state and memory-management information.

Zur Verwaltung von Prozessen muss ein Betriebssystem Informationen über den Prozess speichern.

- Information in a process control block is updated during the transition of process states.
- When the process terminates, its PCB is returned to the pool from which new PCBs are drawn.
- Each process has a single PCB.

Welche Arten von Prozessverwaltung gibt es::https://www.studytonight.com/operating-system/cpu-scheduling
Long Term Scheduler Long term scheduler runs less frequently
Short Term Scheduler This is also known as CPU Scheduler and runs very frequently.
Medium Term Scheduler This scheduler removes the processes from memory

1. Non-Preemptive Scheduling : Under non-preemptive scheduling, once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or by switching to the waiting state.
2. Preemptive Scheduling: the tasks are usually assigned with priorities. At times it is necessary to run a certain task that has a higher priority before another task although it is running.

Nenne und erkläre grundlegende Algorithmen der Prozessverwaltung!

1. First Come First Serve(FCFS) Scheduling:
   the process which requests the CPU first, gets the CPU allocated first. using a Queue data structure, where a new process enters through the **tail** of the queue, and the scheduler selects process from the **head** of the queue.

2. Shortest-Job-First(SJF) Scheduling:

   works on the process with the shortest **burst time** or **duration** first. It is of two types:

   1.Non Pre-emptive
   2.Pre-emptive

   •To successfully implement it, the burst time/duration time of the processes should be known to the processor in advance, which is practically not feasible all the time.

   •This scheduling algorithm is optimal if all the jobs/processes are available at the same time.

5. Priority Scheduling:

   the larger the burst time the lower is the priority of that process.

   Processes with same priority are executed in FCFS manner

   Preemptive Priority Scheduling: If the new process arrived at the ready queue has a higher priority than the currently running process,

   Non-Preemptive Priority Scheduling: if a new process arrives with a higher priority than the current running process, the incoming process is put at the head of the ready queue,

   which means after the execution of the current process it will be processed.

6. Round Robin(RR) Scheduling:

   A fixed time is allotted to each process, called **quantum**, for execution.

   •Once a process is executed for given time period that process is preemptied and other process executes for given time period.

   •Context switching is used to save states of preemptied processes.

9. Multilevel Queue Scheduling

A multi-level queue scheduling algorithm partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of the process, such as memory size, process priority, or process type. Each queue has its own scheduling algorithm.

10. Multilevel Feedback Queue Scheduling:

In a multilevel queue-scheduling algorithm, processes are permanently assigned to a queue on entry to the system. Processes do not move between queues. This setup has the advantage of low scheduling overhead, but the disadvantage of being inflexible.

Was ist eine Ready-Liste und wozu dient sie?

The ready processes in the system are kept in a `queue' called the **ready list**. This list is sorted by the priority of the processes; the lowest priority process appears at the head of the list and the highest priority process is at the tail. Processes of the same priority are sorted by the order in which they are to get service.

- Müssen Prozesse beendet werden? Wenn ja, wann und wie? Wenn nein, warum nicht? Wie lange existieren Prozesse dann?