Echtzeitsysteme Klausurfragen:

1. Was ist ein End-to-End Protokoll?
   Wofür ist es wichtig?

-steuert und überwacht die beabsichtigte Wirkung einer Kommunikation an den Endpunkten
-semantisches Feedback auf der Applikations Ebene
-Hohe Abdeckung an auftretenden Fehlern, durch Überprüfung vom Zustand
Beispiel: Ansteuerung von einem Ventil mit Auslesen vom Zustand

2. Was versteht man unter einem G-State?
   Wofür ist das wichtig?

   -ist der ground state von einem System
   -ist ein minimaler h-state
   -Alle Tasks sind inaktiv und alle Kommunikationskanäle sind geflusht
   -dient als Wiedereingliederungspunkt

3. Was ist ein Agreement Protokoll?
   Wofür ist das wichtig?

   -liefert einen Konsens auf einen Wert einer Beobachtung und auf die Zeit wenn die Beobachtung innerhalb einer Gruppe von fehlerfreien Mitgliedern durchgeführt wurde
   -Wann immer eine dichte Menge auf eine diskrete Repräsentation abgebildet wird, werden Agreement Protokolle benötigt um identische Inputs in ein System zu geben um identische Outputs zu erwarten

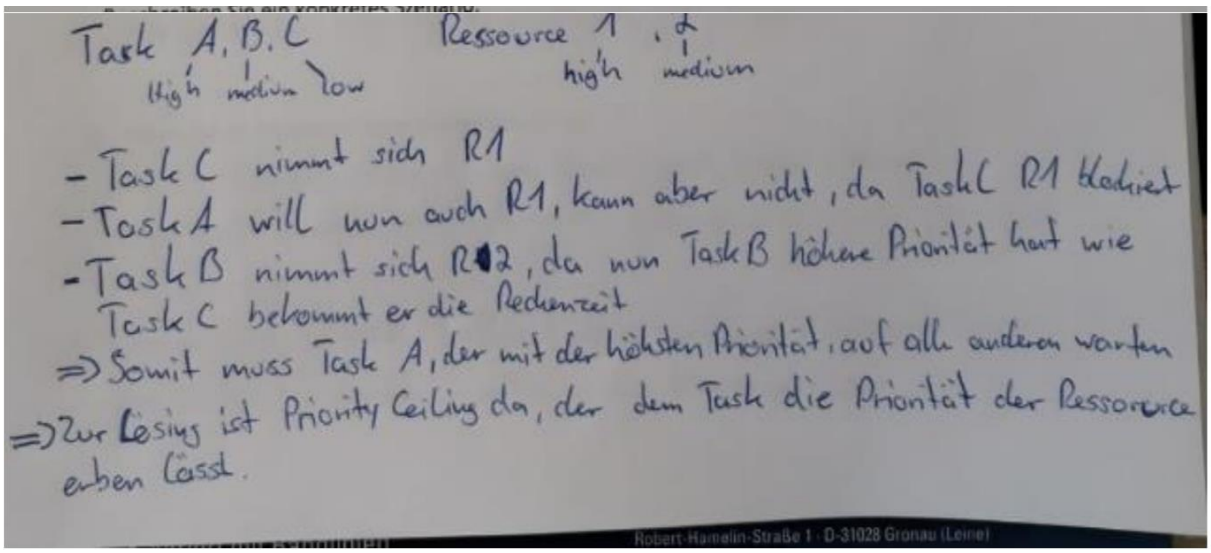
4. Was ist Priority Ceiling?
   Unterschied zu Priority Inheritance?
   Wofür genau wird es verwendet?
   Beschreiben Sie ein konkretes Szenario.

   -Priority Ceiling:
      -Schedulingverfahren, welches Deadlocks und Priority Inversion verhindert
      - jeder Prozess hat eine Default Priorität
      -zu jeder Resource wird eine Prioritäts Schranke (Ceiling) festgelegt
      -zu jedem Zeitpunkt wo ein Task ausgeführt wird bekommt er eine dynamische Priorität, welche dem Maximum seines eigenen statischen Priorität und dem Ceiling Wert von allen Ressourcen die er geloggt hat entspricht
      -> ein Task darf nur dann eine neue Ressource aufnehmen, wenn seine Priorität höher ist als die Ceilings von allen Resourcen die durch andere Tasks geloggt sind

   Unterschied zu Priority Inheritance:
      - Priority Inheritance kann keine Deadlocks verhindern

   Verwendung:
      -Wird vom Scheduler benutzt damit jeder Task ordnungsgemäß ausgeführt werden kann

   Konkretes Szenario:



5. Wann ist eine Globale Zeitbasis reasonable?
   Warum ist das wichtig?

   Eine globale Zeitbasis ist reasonable wenn die Granularität größer als die Präzision ist: $\Pi < g$

   -> Der Synchronizationsfehler ist kleiner als ein Macrogranule (Makrotick der globalen Zeit)

6. Wie sind die Begriffe Accuracy und Precision definiert?
   Wie ist die Precision des Central Master Algorithmus?

   Accuracy: Ist der Offset einer Uhr und der Referenzuhr zu einem Tick i
   Precision: Ist der Offset zwischen zwei Uhren zu einem bestimmten Zeitpunkt (Tick i)

   Precision CMA:
   - Precision des CMA (Π)= Latency Jitter (ϵ) + Drift Offset ($\Gamma = 2 \rho R_{int}$)

   $$\Pi_{central} = \epsilon + \Gamma$$

7. Nennen Sie drei Beispiele für „a priori“ Wissen, welches für die Erfüllung von Echtzeitanforderungen verwendet werden kann.

   -A priori Wissen über des Verhalten wird benutzt um Fehlererkennung zu erhöhen, denn man weiß wann ein Knoten eine Nachricht senden muss
   -Es wird kein Identifier mehr benötigt, denn der Zeitpunkt an dem eine Nachricht gesendet wurde identifiziert den Absender
   -Man weiß wie viele Nachrichten in einem Hochlast Szenario ankommen, daduch können Ressourcen danach geplant werden

8. Nennen und begrüden Sie die fundamentalen Grenzen der Zeitmessung.

   -Wenn ein einzelnes Event von zwei Knoten beobachtet wird, können die lokalen Timestamps für dieses Event um 1 Tick abweichen

   -Eine Zeitmessung kann +-2 Ticks der globalen Zeit von der tatsächlichen Dauer abweichen

   -Zwei Timestamps müssen mindestens 2 Ticks auseinander sein um die Reihenfolge bestimmen zu können

   -Die zeitliche Ordnung von einer Menge von Ereignissen kann immer abgeleitet werden, wenn die 0/3g-precedent gesetzt ist

   Begründung:
   -da ein Event jeder Zeit auftretten kann und es deshalb immer möglich ist, dass es zwischen zwei lokalen Timestamps der Uhren liegt
   -ein Event kann mehrere Zeitstempel besitzen und so verschieden aufgefasst werden
   ->dies tritt auf da es immer eine gewisse Precision gibt also einen Offset zwischen den zwei Uhren zu einem bestimmten Zeitpunkt

9. Erklären Sie das Adversary Argument.

   Ein sporadischer Task muss dran kommen sobald er dran kommen will, wenn die Worst Case Execution Time der Deadline ist. Ansonsten verpasst er die Deadline.
   Wenn das Wissen über die Zukunft nicht bekannt ist, wann ein sporadischer Task dran kommen möchte, kann man keine Lösung finden. Das ist das Adversary Argument. Obwohl es eine Lösung gibt kann der Online Scheduler diese nicht finden, da er nicht weiß das der sporadische Task kommt.

10. Was versteht man unter State-Estimation?
    Geben Sie ein Beispiel an?
    Beschreiben Sie einen möglichen Zusammenhang zur Uhrensynchronisation?

    -Eine Vorausschau wie sich das Verhalten ändert
    -Vorhersagen von Werten
    -man arbeitet mit Werten aus der Vergangenheit und versucht damit auf zukünftige Werte zu schließen

    Beispiel: Temperatur Sensor liefert nur alle 10 sec einen Wert, doch das System möchte jede Sekunde einen Wert für die Ausgabe haben => es werden Werte aus der Vergangenheit genommen um auf die Werte der Zukunft / aktuellen Wert zu schließen.

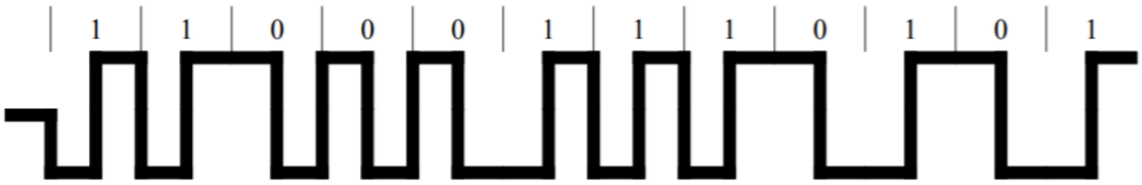
    - wenn man aus den alten Daten weiß, dass man immer um ca. 1 sek von der Refrenzuhr abweicht, kann man sich selbst um 0.01 sek zurückstellen um wieder gleich mit der Referenzuhr zu sein. Und damit die Abweichung bei der nächsten Synchronisation nicht so hoch ist.

11. Geben Sie ein Beispiel für einen Hidden Channel an. In welchem Zusammenhang steht dazu der Begriff Permanence?

    Im System, z.B. Druckkessel, gibt es einen Sensor für den Druck und einen Aktuator, der das Ventil zum Druckablass bedient. Wenn man das System beeinflussen möchte, man man dem Aktuator sagen, dass er den Druck ablassen soll. Der Aktuator lässt darufhin den Druck ab, ohne dem Sensor zu benachrichtigen. Der Sensor misst aber den Druck und erkennt eine Systemveränderung. Der Hidden Channel ist in diesem Fall die Verbindung zwischen dem Aktuator und dem Sensor.

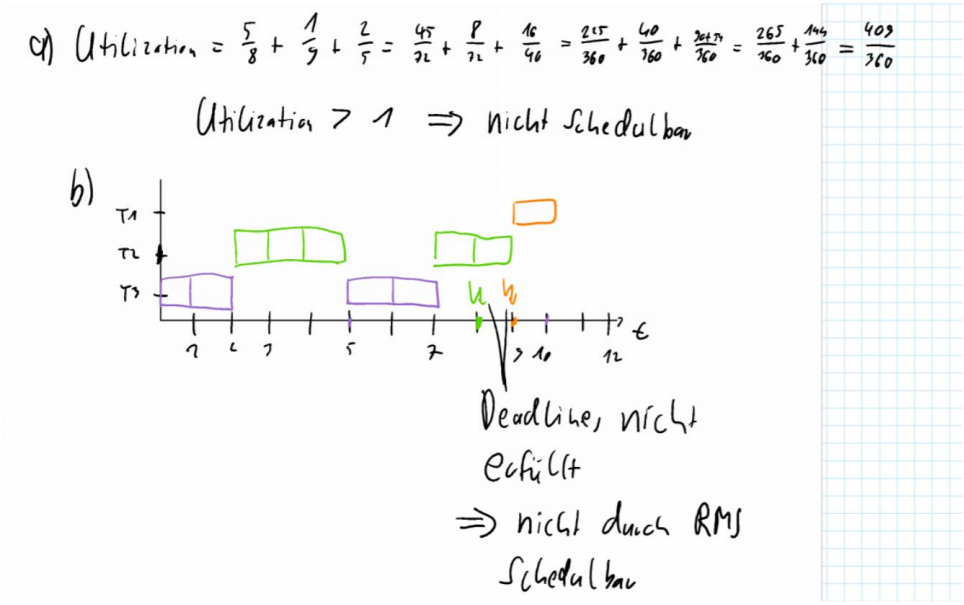
    Permanence:
    Eine Nachricht wird permanent sobald alle Nachrichten die vorher an dieses gleiche Objekt gesendet worden sind angekommen sind. Abgeleitete Aktionen von Nachrichten die nicht permanent sind können zu Fehlern und inkonsistenz führen. Z.B. Fehlalarm bei zu hohem Druck obwohl der Druck durch einen Operator schon abgelassen wurde und diese Nachricht aber noch nicht beim Alarm angekommen ist

Das nicht 12.Codieren Sie die Bitfolge 1000010011010 mit dem Manchester Code



- 1 flankenwechsel und 0 nicht. bie 00 flankenwechsel zum clockzeitpunkt (clockbit)

13.Gegeben sind folgende unabhängige periodische Tasks, mit Deadline Interval gleich Taskperiode: T1(5,8); T2(1,9); T3(2,5)
-ermitteln Sie mit dem Schedulability Test ob dieses Taskset mit dem Rate Monotonic Scheduling Verfahren auf einem Prozessor schedulbar ist.
-zeigen Sie anhand des konkreten Schedules ob dieses Taskset mit dem Rate Monotonic Scheduling Verfahren auf einem Prozessor schedulbar ist



14. Zählen Sie drei wichtige Eigenschaften von Leitungscodes auf.

-Elektromagnetische Verhalten
- Bandbreite
- Taktrückgewinnung

Nicht ihn VL

15. Was ist bitstuffing?
Wofür wird es benötigt?

-Bitstuffing ist wenn bei langen unveränderten Signalen nach einier definierten Menge von (gleichen) übermittelten Werten (bei NRZ nach jedem 5 Bit) eine inverses Bit eingefügt wird, sodass die Taktrückgewinnung ermöglicht wird. So wird verhindert, dass die Uhr des Empfängers und die des Senders zu stark auseineander driften. Aber erhöht Protokoll Latency Jitter.

16. Nennen Sie die drei Timeouts des ARINC 629 Protokolls.
Beschreiben Sie deren Zweck.

- Synchronization Gap: Kontrolliert den Eingang zum Wartezimmer
- Terminal Gap: Kontrolliert den Zugang zum Bus
- Transmit Interval: Deaktiviert die Benutzer um deren alleinige Nutzung des Busses zu verhindern

Nicht ihn VL

17. Wie wird bei CAN der Zugriff auf das Medium geregelt?
Wodurch ist die maximale Übertragungsrate bestimmt?

Bei CAN wird der Zugriff auf das Medium mittels einen dominanten Signalpegel (Bit) geregelt. Jeder Knoten legt seinen Identifier auf den Bus. Der Knoten, welcher die dominanteren Signalpegel (Bit) besitzt, darf seine Nachricht zu Ende senden. Die anderen nicht dominanten Knoten hören dabei auf zu senden.

Durch die Ausbreitungsgeschwindigkeit der Bits auf der Leitungsstrecke und durch die Länge der Leitung. Da ein Knoten bei jedem Bit überprüfen muss ob sein eigenes Signal auf dem Bus liegt muss er warten bis evtl. ein dominantes Signal von der anderen Seite des Busses zu ihm kommt bevor er weiter senden kann

18. Welches Media Access Control Verfahren wird bei CAN verwendet (CSMA/CA oder CSMA/CD)?
Begründung?

CSMA/CA: Kollisionen beim Buszugriff werden durch die Arbitrierung aufgelöst
-Bei CAN soll die Information schnell weitergeleitet werden
-Das dominante Bit gewinnt und kann sofort weitersenden

19. Was ist Priority Inheritance?
Wofür wird es verwendet?
Beschreiben Sie ein konkretes Szenario.

- Priority Inheritance verhindert Priority Inversion, indem ein blockierender Task die höchste Priorität aller von ihm blockierten Tasks erbt.
- Wenn ein Task mit niedriger Priorität zuerst eine Ressource blockiert, danach ein Task mit höherer Priorität die Ressource möchte und darauf warten muss, dann aber ein Task mit mittlerer Priorität kommt und dem 1. Task die Rechenzeit stielt ist das Priority Inversion, weil ein Task mit höherer Priorität nun auf alle Tasks mit höherer Priorität als der 1. Task warten muss.
  Priority Inheritance löst dieses Problem indem es dem 1. Task sobald der Task mit hoher Priorität auf die Ressource zugreifen will, für die Dauer des Locks die hohe Priorität gibt. Nun kann der 1. Task ununterbrochen von dem Tasks mit mittlerer Priorität seine Arbeit beenden.

20. Nennen Sie die Größe des Action Delays in Systemen ohne globale Zeit.

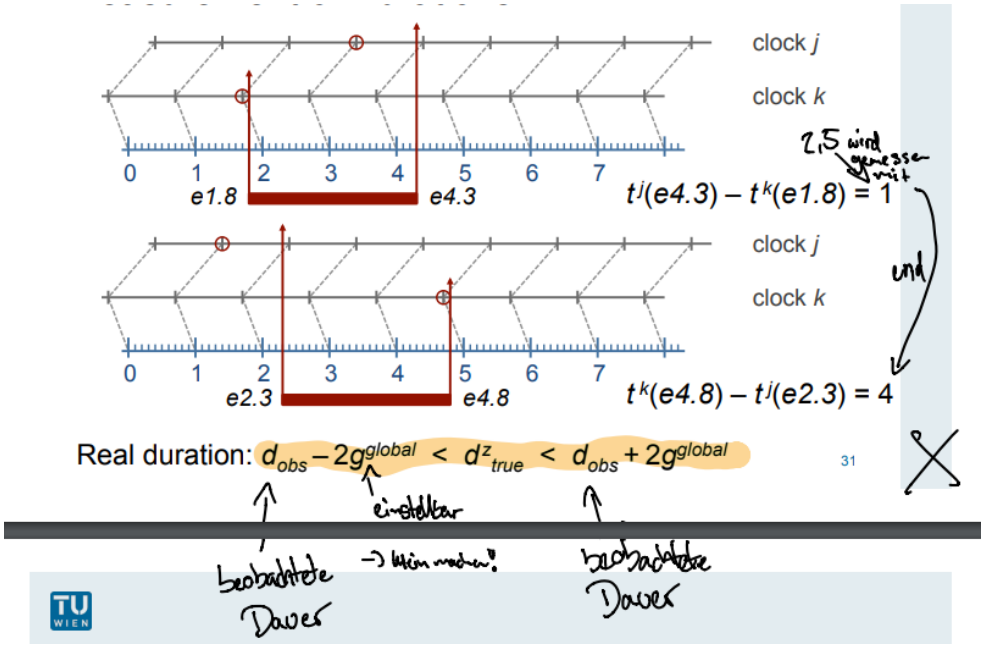
Send Time of Message (t_{send}), Maximum Communication Delay (d_{max}), Minimum Communication Delay (d_{min}), Lokale Granularität (g_l), Globale Granularität (g)

- Action Delay ohne globale Zeit:

$$t_{permanent} = t_{send} + 2d_{max} - d_{min} + g_l \quad \text{(von außen)}$$
$$d_{action} = t_{permanent} - t_{send} = d_{max} + \epsilon = 2d_{max} - d_{min} \quad \text{(von innen)}$$

21. Wie genau können Intervallgrenzen mit globaler Zeit gemessen werden? Geben Sie ein Beispiel an.

- jede Grenze für sich kann um g genau gemessen werden, daraus kann Man für die Dauer des Intervalls schließen: (d obs – 2g) <= d real <= d obs + 2g)



22. Wie funtioniert der EDF-Algorithmus, wie der Least Laxity Algorithmus? Welcher ist besser?

EDF:
-der Task mit der frühesten Deadline bekommt die höchste Priorität
-absolute deadlines bestimmen die Ausführungsreihenfolge der Tasks

Leas Laxity:
-der Task mit der kleinsten Differenz zwischen Deadline und der übrigen Rechenzeit (Laxity) bekommt die höchste Priorität

-> bei einem Prozessor sind beide Algorithmen optimal, bei mehreren Prozessoren ist der Last Laxity jedoch besser da er Schedules auch dort findet wo EDF keine findet

23. Weisen Sie nach, dass es keinen optimalen Online-Scheduler gibt.

Ein Online Scheduler kann nicht in die Zukunft schauen und so zufällig auftrettende Task nicht einplanen (Adversary Argument).

Ein sporadischer Task muss dran kommen sobald er dran kommen will, wenn die Worst Case Execution Time der Deadline ist. Ansonsten verpasst er die Deadline.
Wenn das Wissen über die Zukunft nicht bekannt ist, wann ein sporadischer Task dran kommen möchte, kann man keine Lösung finden. Das ist das Adversary Argument. Obwohl es eine Lösung gibt kann der Online Scheduler diese nicht finden, da er nicht weiß das der sporadische Task kommt.

24. Erläutern Sie zwei Arten der Uhrensynchronisation.
Wofür werden sie verwendet?

- Interne Uhrensychronisation: Alle Uhren eines geschlossenen Systems gleichen ihre Zeit untereinander ab. Danach gehen alle Uhren mit einer gewissen Precision gleich.
- Externe Uhrensychronisation: Die Uhr gleicht sich mit einem externem System ab. Dieses System gibt eine Zeit vor, welche angenommen wird, ohne „Mitspracherecht".

Interne Uhrensynchonisation: hohe Verfügbarkeit, gute Kurzzeit Stabilität
Externe Uhrensynchronisation: Langzeit Stabilität, mögliche schlechtere Verfügbarkeit

25. Was ist der Unterschied zwischen Sampling und Polling?
Erläutern Sie die Vor- bzw. Nachteile.

- Vom funktionellen Aspekt her gibt es keinen Unterschied. Nur der Aufbau ist unterschiedlich. Beim Sampling liegt das Memory-Element in der Sphere of Control des Sensors, beim Polling in dem vom Controller.
- Im Fehlerzustand ist Sampling robuster, da wenn die Übertragungsleitung gestört wird, dies nur Einfluss auf den Zeitpunkt des Abtastens hat. Beim Polling merkt sich das Memory-Element jeden Fehler auf der Übertragungsleitung, das kann leichter zu fehlerhaft abgetasteten Werten führen.
- Wenn der Controller neugestartet wird sind beim Polling alle Werte im Memory-Element gelöscht, da es in der SOC liegt. Beim Sampling bleiben die Werte darin weiterhin vorhanden.

??????????????????????????????

26. Gegeben sind: Bandbreite 2 Mbit/s, Kanal 2 km Länge, Nachrichtenlänge 320 Bit.
Geben Sie die Grenze der Protokolleffizienz an.

$$ProtocolEffeciency = \frac{MessageLength}{MessageLength + BitLength}$$

$$BitLength = PropagationDelay \cdot Bandwidth$$

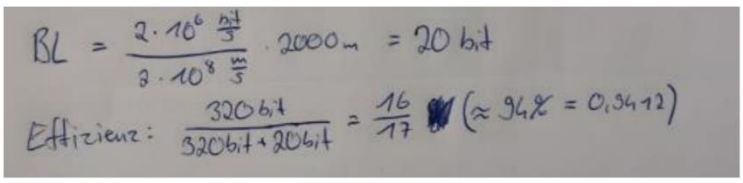
$$PropagationDelay = \frac{CableLength}{SpeedThroughCable}$$

$$SpeedThroughCable = 200 \cdot 10^6 \frac{m}{s}$$

$$PropagationDelay = \frac{2000\,m}{200 \cdot 10^6 \frac{m}{s}} = 10^{-5}\,s$$

$$BitLength = 10^{-5}\,s \cdot 2 \cdot 10^6 \frac{Bit}{s} = 20\,Bit$$

$$ProtocolEffeciency = \frac{320\,Bit}{320\,Bit + 20\,Bit} = 0.94 = 94\,\%$$

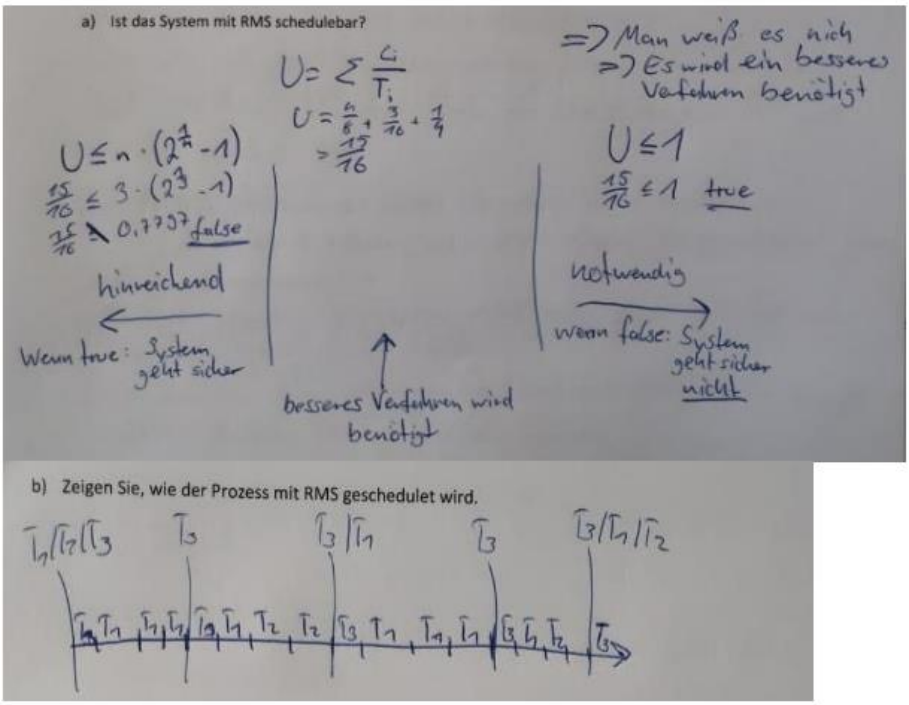


27. Welche Arten der Flusskontrolle kennen Sie, welchen Einfluss haben dies auf das Echtzeitverhalten Ihres Systems?

- Explizite Flusskontrolle: Sender kann immer senden wenn der Kanal frei ist. Empfänger sendet nach jeder empfangenen Nachricht eine Bestätigung. Dadurch wird Rückkanal notwendig. SOC liegt beim Empfänger, Fehlererkennung beim Sender. Kann bei großer Last Trashing verursachen, dies führt zu einem größeren Protokoll Latenz Jitter und ist daher für Echtzeitsysteme eher ungeeignet.
- Implizite Flusskontrolle: Sender sendet in vorher fest abgemachten Intervallen Nachrichten. Wenn der Empfänger keine Nachricht bekommt, muss er darauf reagieren, daher liegt die Fehlererkennung beim Empfänger. Antwort des Empfängers ist nicht notwendig, daher auch Rückkanal nicht notwendig. Multicast sehr einfach realisierbar. Wenn richtig geplant auch bei großer Last noch funktionsfähig. Kleiner Protokoll Latenz Jitter, daher für RTS geeignet.

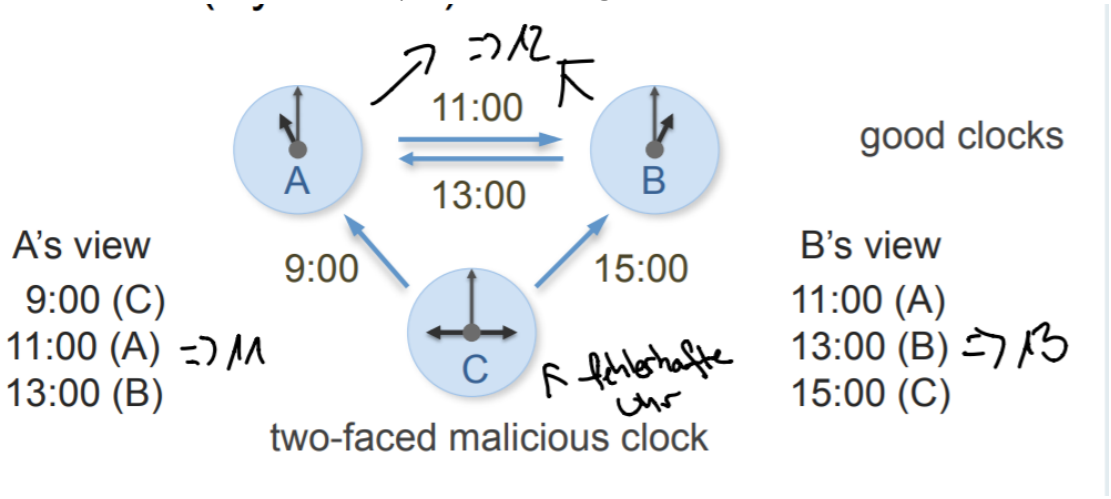
28. Gegeben sei folgendes System:
    - ist das System mit RMS schedulebar?
    -Zeigen Sie, wie der Prozess mit RMS geschedulet wird.

	Ci	Ti
T1	4	8
T2	3	16
T3	1	4



29. Was versteht man unter dem byzantischen Fehler?
Inwieweit spielt dieser eine Rolle?

-eine Uhr gibt bei zwei unterschiedlichen Zeitabfragen zwei unterschiedliche und falsche Zeitstände heraus (starke Abweichung)
-dadurch verhält sich das System beliebig falsch



=> man benötigt N >= 3k+1 Uhren, wobei N die Anzahl an Uhren und k die Anzahl der byzantischen Uhren ist
(dadurch kann dann eine Aussage über die aktuelle Zeit gemacht werden, da kein Schaden angerichtet wird)

30. Welchen Schedulablility Algorithmus verwendet RMA?
Erklären Sie Ihre Variablen.

->RMS = Rate-Monotonic-Scheduling
-Prioritäten werden anhand der Periodendauer eines Jobs festgelegt
-je kürzer die Periodendauer eines Jobs, desto höher ist die Priorität
- Ci = Computation Time
- Ti = Periode
- n = Anzahl an Tasks

31. Was ist ein Real Time Image? Welche Aussagen lassen sich treffen?

-Abbild einer Real-Time-Entity in einem System
-gültig an einem bestimmten Zeitpunkt, wenn es die genaue Darstellung der entsprechenden RT-Entity im Wert- und Zeitbereich ist
->ist nur in einem bestimmten Zeitbereich gültig

32. Was ist der h-State?

- History State
- Status eines Knotens zu einer bestimmten Zeit mit bestimmten Werten
-ändert sich über die Zeit
- dort stehen alle Informationen, die benötigt werden um einen intialisierten Knoten oder Task zu einem beliebigen Zeitpunkt zu starten
-hilft bei der Reintegration eines Knotens nach der Initialisierung

33. Nennen Sie die Größe des Action Delays in Systemen mit globaler Zeit.

Send Time of Message (t_{send}), Maximum Communication Delay (d_{max}), Minimum Communication Delay (d_{min}), Lokale Granularität (g_l), Globale Granularität (g)

– Action Delay mit globale Zeit:

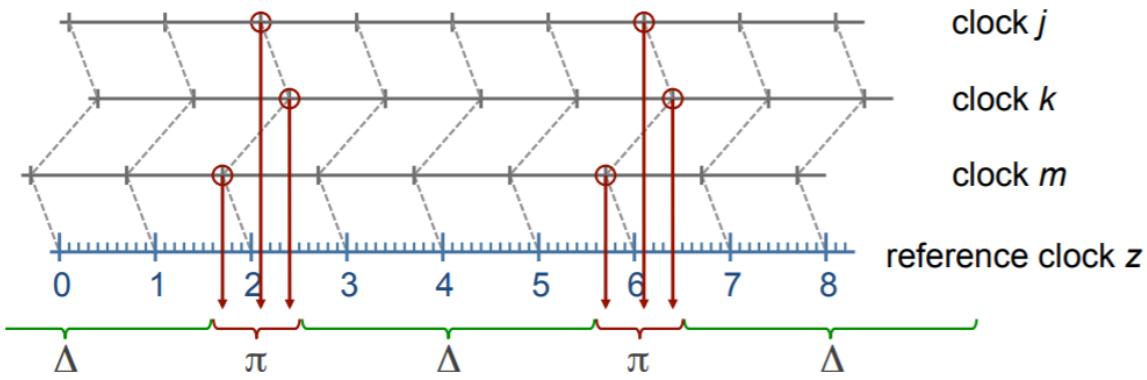
$$d_{action} = t_{permanent} - t_{send} = d_{max} + 2g$$
$$t_{permanent} = t_{send} + d_{max} + 2g$$

34. Was ist externe Uhrensynchronisation? Wofür wird sie verwendet?

- interne Uhr (Uhren) gleicht sich mit einem externen System ab.
- externes System gibt eine Zeit vor, welche ohne Mitspracherecht angenommen wird

->Langzeitstabilität aber womöglich weniger Verfügbarkeit

35.Was ist die pi/delta Präzedenz

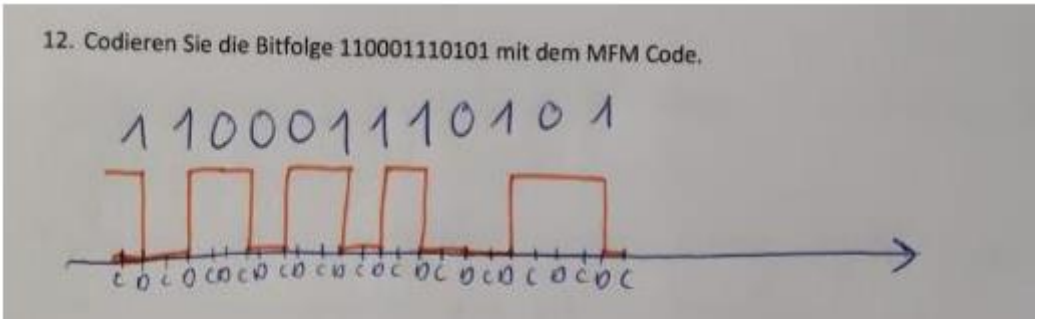


Given durations π and Δ (π << Δ), a set of events $E=\{e_i\}$ is π/Δ-precedent, if the following condition holds for all $e_j, e_k \in E$:

$$(|z(e_j) - z(e_k)| \leq \pi) \text{ or } (|z(e_j) - z(e_k)| > \Delta)$$

33

Nicht in VL



Nicht in VL