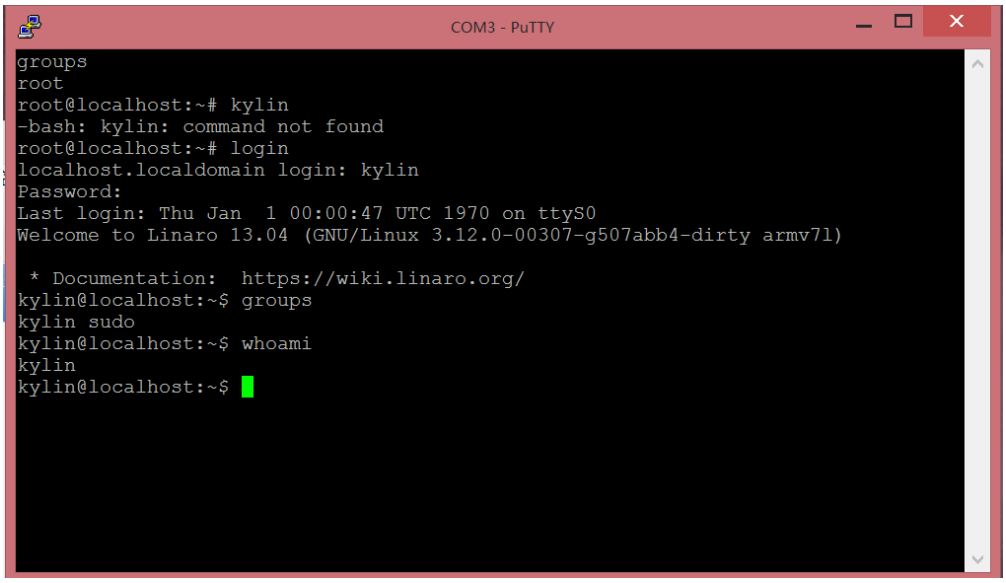
Real Time Embedded Exercise 2
ECEN 5623

Zachary Vogel Chuyi Liu

February 13, 2016

Problem 1

To prove that we did in fact make an account on the DE1 board, We have this screenshot of a login on the board through Putty. The groups command as well as the whoami command illustrate that the account is in fact created.



You can see the login as well as the account confirmation with whoami

Problem 2

Explanation and Critique

Within the frequency executive architecture, off-line processors are applied to compute data and provide a scheduling table. The high-frequency executive is scheduled at a higher priority compared to mid-frequency and low-frequency executive. That is similar to RM scheduling, which helps to ensure the feasibility and safety of the system. It may be applied to ensure all processing is completed in an extremely short time, which is good for systems that have high safety requirements[advantage-1]. Hence, this architecture is very helpful in guidance, navigation, and control(GN&C) software, which performs a wide variety of functions with extremely tight timing and phasing relationships, the execution rates for its principal functions within an unique operation sequence(OPS) vary from 25 to 0.25 Hz. This executive architecture is table-driven, its scheduling method is very deterministic, thus the executives are very simple and repeatable[advantage-2] compared to real-time software system. And thanks to which the system is more predictable[advantage-3] as

long as overruns don?t occur. However, the table is not very flexible, we will have to recompute then change the whole table to make a little change, while in a real-time software system we don?t need to do that, for other words, it?s much harder to implement additional functions or requirements[disadvantage-1]. PASS uses eight different phase/function combinations, and each OPS has different parts, modes, and thus substructures are implemented(like man/machine interface structure). This hierarchical structure makes the computation for the architecture more complex than for a real-time system. Due to the complexity, it requires a lot of off-line computation, which relays on off-line processors[disadvantage-2]. To make a successful implementation of PASS software, the system is supposed to pursue the implementation of software requirements in a top-down sequence and follow a release plan that used a building-blocking approach. Hence, tasks are supposed to have fixed release times[disadvantage-3].

Problem 3

Coding

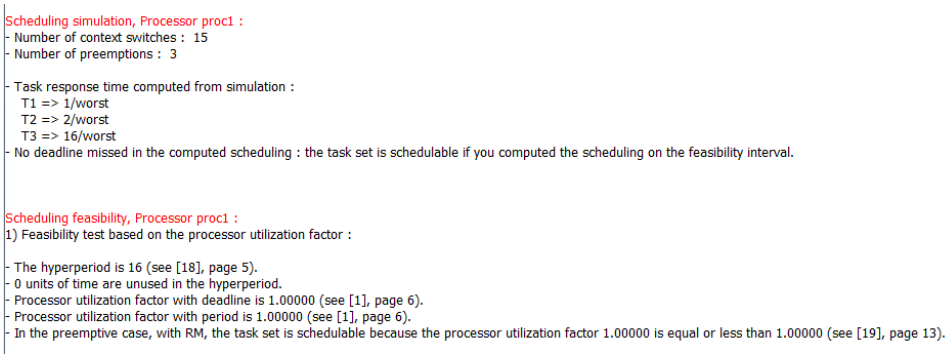
We wrote a scheduler for LLF and EDF. We don't think they work because they pass tests that failed in cheddar, specifically sim 6 and 8 for LLF. We have no idea why, we spent hours debugging with no success. As far as we can tell the implementation gives the correct output. There are some print statements to give the output if you want to check, but they all seem to follow the rules. The only other thing that mgiht be causing a problem is setting the Deadline time to the LCM when computation time is reset. The code is attached with a makefile. We also did a deadline utilization function, which similarly to the RM LUB, won't pass any task sets that aren't feasible. It will fail some tests that are though. Both schedulers use the same idea, find min deadline or laxity, decrement computation time, always decrement the time to deadline, exit when the remaining computation is greater than deadline time, and reset on period. In that order specifically.

As for the feasibility tests, we could not determine a simple computation that gave a necessary and sufficient answer to the scheduling problem. We would be interested in seeing how Cheddar does this because most of the papers we read did not give a feasibility test outside of pure scheduling. Cheddar seems to do some calculation of worst case execution time, but it doesn't seem trivial to calculate.

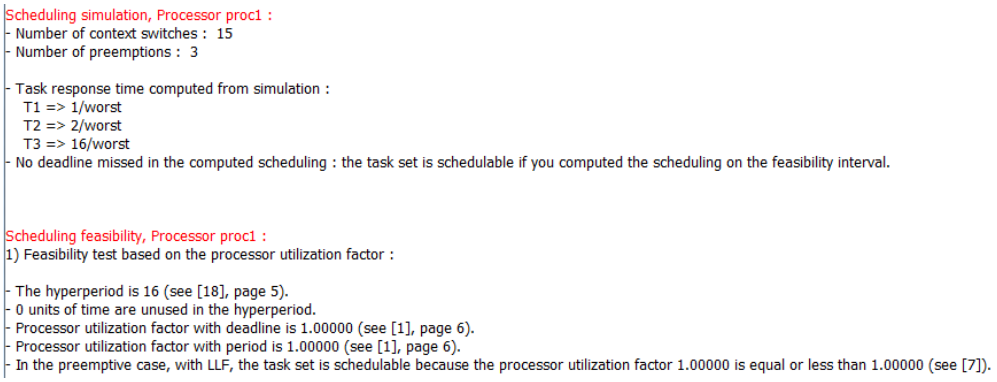
Cheddar Conformation

Here we have all of the output from cheddar for the various problems.

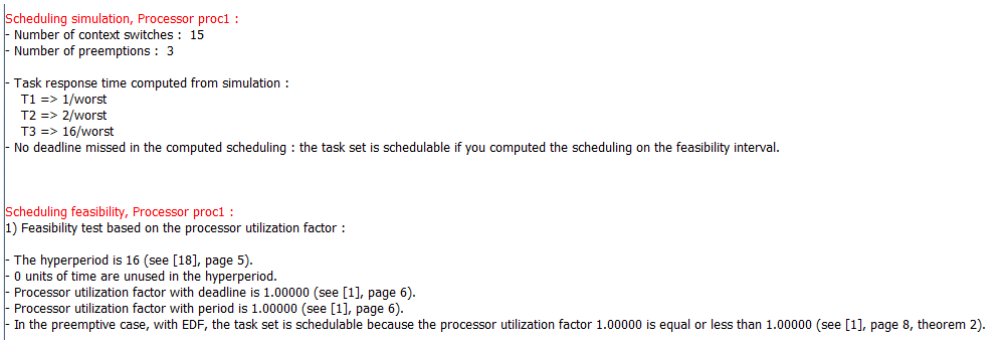
Example 4



Example 4 Rate Monotonic

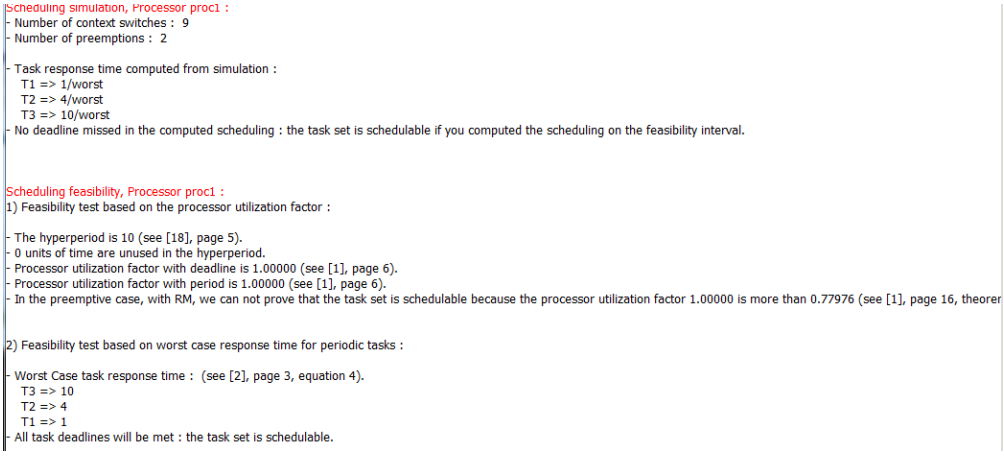


Example 4 Least Laxity First



Example 4 Earliest Deadline First

Example 5



Example 5 Rate Monotonic

Scheduling feasibility, Processor proc1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 10 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
  T1 => 2
  T2 => 5
  T3 => 10
- All task deadlines will be met : the task set is schedulable.

Scheduling simulation, Processor proc1 :
- Number of context switches : 9
- Number of preemptions : 2

- Task response time computed from simulation :
  T1 => 1/worst
  T2 => 4/worst
  T3 => 10/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Example 5 Least Laxity First

Scheduling simulation, Processor proc1 :
- Number of context switches : 9
- Number of preemptions : 2

- Task response time computed from simulation :
  T1 => 1/worst
  T2 => 4/worst
  T3 => 10/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Scheduling feasibility, Processor proc1 :
1) Feasibility test based on the processor utilization factor :

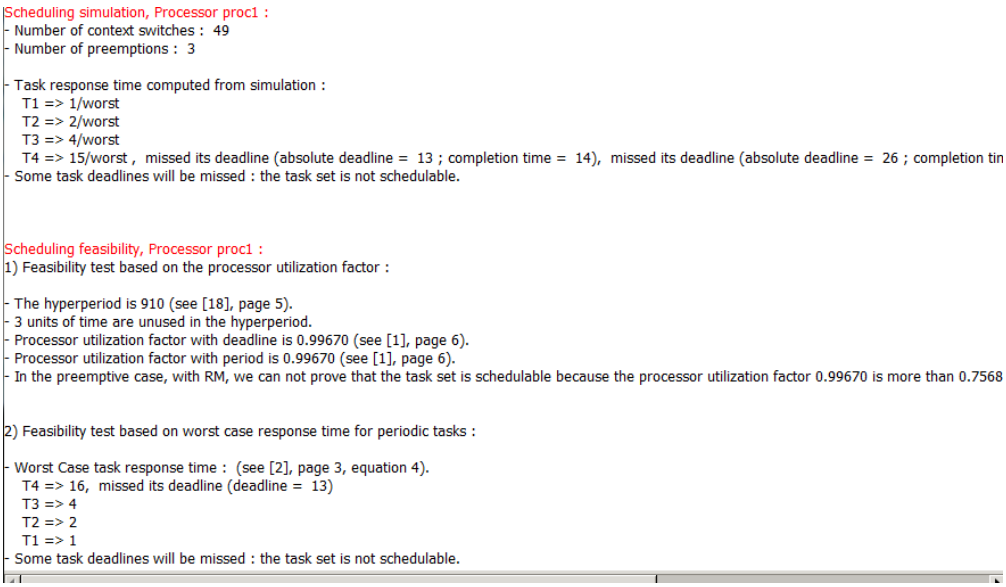
- The hyperperiod is 10 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [1], page 8, theorem 2).

2) Feasibility test based on worst case response time for periodic tasks :

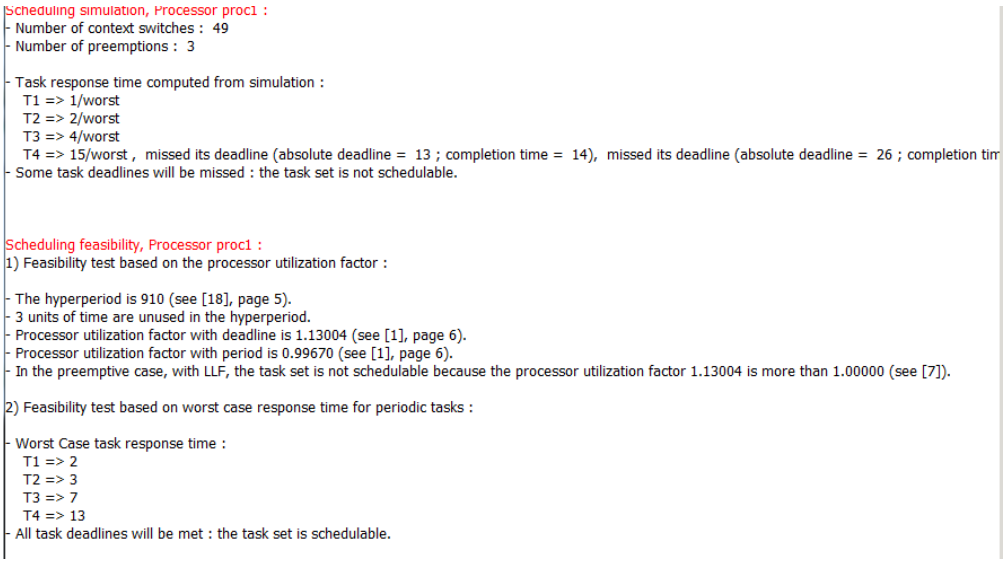
- Worst Case task response time :
  T1 => 2
  T2 => 5
  T3 => 10
- All task deadlines will be met : the task set is schedulable.

Example 5 Earliest Deadline First

Example 6



Example 6 Rate Monotonic



Example 6 Least Laxity First

```
Scheduling simulation, Processor proc1 :
- Number of context switches :  49
- Number of preemptions :  4

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 2/worst
   T3 => 7/worst
   T4 => 12/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.



Scheduling feasibility, Processor proc1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.13004 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with EDF, the task set is not schedulable because the processor utilization factor 1.13004 is more than 1.00000 (see [1], page 8,


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 2
   T2 => 3
   T3 => 7
   T4 => 13
- All task deadlines will be met : the task set is schedulable.
```

Example 6 Earliest Deadline First

Example 7

```
Scheduling simulation, Processor proc1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 3/worst
   T3 => 15/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.



Scheduling feasibility, Processor proc1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 15 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 1.00000 is more than 0.7797


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
   T3 => 15
   T2 => 3
   T1 => 1
- All task deadlines will be met : the task set is schedulable.
```

Example 7 Rate Monotonic

```
Scheduling simulation, Processor proc1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 3/worst
   T3 => 15/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor proc1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 15 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 3
   T2 => 5
   T3 => 15
- All task deadlines will be met : the task set is schedulable.
```

Example 7 Least Laxity First

```
Scheduling simulation, Processor proc1 :
- Number of context switches :  11
- Number of preemptions :  3

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 3/worst
   T3 => 15/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor proc1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 15 (see [18], page 5).
- 0 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 1.00000 (see [1], page 6).
- Processor utilization factor with period is 1.00000 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 1.00000 is equal or less than 1.00000 (see [1], pag


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 3
   T2 => 5
   T3 => 15
- All task deadlines will be met : the task set is schedulable.
```

Example 7 Earliest Deadline First

Example 8

```
Scheduling simulation, Processor proc1 :
- Number of context switches :  14
- Number of preemptions :  1

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 2/worst
   T3 => 4/worst
   T4 => 14/worst ,  missed its deadline (absolute deadline =  13 ; completion time =  14)
- Some task deadlines will be missed : the task set is not schedulable.



Scheduling feasibility, Processor proc1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with RM, we can not prove that the task set is schedulable because the processor utilization factor 0.99670 is more than 0.7568


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :  (see [2], page 3, equation 4).
   T4 => 16,  missed its deadline (deadline =  13)
   T3 => 4
   T2 => 2
   T1 => 1
- Some task deadlines will be missed : the task set is not schedulable.
```

Example 8 Rate Monotonic

```
Scheduling simulation, Processor proc1 :
- Number of context switches :  14
- Number of preemptions :  1

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 2/worst
   T3 => 4/worst
   T4 => 14/worst ,  missed its deadline (absolute deadline =  13 ; completion time =  14)
- Some task deadlines will be missed : the task set is not schedulable.



Scheduling feasibility, Processor proc1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with LLF, the task set is schedulable because the processor utilization factor 0.99670 is equal or less than 1.00000 (see [7]).

2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 1
   T2 => 4
   T3 => 6
   T4 => 12
- All task deadlines will be met : the task set is schedulable.
```

Example 8 Least Laxity First

```
Scheduling simulation, Processor proc1 :
- Number of context switches :  14
- Number of preemptions :  1

- Task response time computed from simulation :
   T1 => 1/worst
   T2 => 4/worst
   T3 => 5/worst
   T4 => 10/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling feasibility, Processor proc1 :
1) Feasibility test based on the processor utilization factor :

- The hyperperiod is 910 (see [18], page 5).
- 3 units of time are unused in the hyperperiod.
- Processor utilization factor with deadline is 0.99670 (see [1], page 6).
- Processor utilization factor with period is 0.99670 (see [1], page 6).
- In the preemptive case, with EDF, the task set is schedulable because the processor utilization factor 0.99670 is equal or less than 1.00000 (see [1], p


2) Feasibility test based on worst case response time for periodic tasks :

- Worst Case task response time :
   T1 => 1
   T2 => 4
   T3 => 6
   T4 => 12
- All task deadlines will be met : the task set is schedulable.
```

Example 8 Earliest Deadline First

Problem 4

Constraints

- The deadline of a system is equivalent to its period.
- Fixed Priority, Preemptive, Run-to-Completion Scheduling is used
- The LUB must never pass a task set that will fail

Assumptions

These are largely the same assumptions they make for the rate monotonic protocol.

- The requests for all tasks with hard deadlines are periodic with constant interval between each request
- Each request must be capable of finishing within its period, this is known as the run-ability constraint
- Run-time is known and deterministic for these evaluations.

Key steps

The important thing to note about this derivation is that it starts with a set of rules that works under a very strict set of constraints and then works towards being more general. The major theorems that are used all try to generalize previous steps to a broader set of constraints.

- One of the key steps for the LUB derivation is providing a priority assignment. This happens before the section on the derivation, but is essential to the proof because it provides an assignment of priorities. The proof for this is done on a set of two tasks, but the way it is done could be extended to infinitely many more.
- Another essential step is Theorem 3. It defines the least upper bound for a set of two tasks, this is important because it provides a mathematical basis for extending the bound to multiple tasks. It follows the idea that you can start simply and then add complexity as you go. The proof is also kind of "tricky" because it requires the analysis of two different cases that a two task system could be in. The two cases are when task 1 finishes quickly enough to not effect task 2's critical section and when it doesn't.

- Theorem 4 adds another level of complexity by extending the bound to m tasks, while adding in the restriction that the ratio between request periods is less than 2. I consider this interesting because the Liu and Layland used the ratio between periods to find an effective way to bridge the proof gap between Theorem 3 and Theorem 5. It is also nice that they use derivatives to find the minimum towards the end of that proof.