

R.T.S Assignment Report

Real Time System



November 2, 2015

Ren kai tam

R00095982

Contents

[Problem 1.1 3](#_Toc436039839)

[Results: 3](#_Toc436039840)

[Problem 1.2 3](#_Toc436039841)

[Results: 3](#_Toc436039842)

[Problem 1.4 4](#_Toc436039843)

[Results: 4](#_Toc436039844)

[Problem 1.5 – Self Driving Car (Smart Car) 5](#_Toc436039845)

[Results: 6](#_Toc436039846)

# Problem 1.1

The first problem contain 4 processes, startup and 3 other processes.

***P2***

Startup program forks 3 processes. Process 2 and 3 takes in a sets of parameters.

***P1***

***Startup***

* P1 prints out 3 OS-9 system call messages. Each message delayed for 1.5 seconds before displaying.

{"P3", "3", "-", "2", 0}

{"P2", "A", "AB", "ABC", 0}

* P2 expecting a list of string and then it calculate the length of the string.

***P3***

* P3 expecting a string that contain 2 numbers and a mathematical operator [i.e.] ‘+’, ‘-’, ‘\*’, ‘/’

## Results:

(INSERT IMAGE HERE)

# Problem 1.2

To solve this problem instead of using parameter passing to each process from the startup, an alarm cycle us used during the process.

***P1***

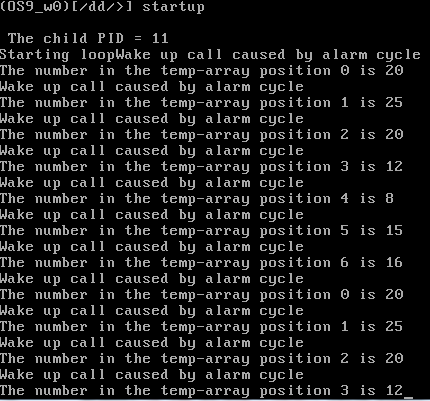
***Startup***

Startup fork a process, and it doesn’t pass in any parameters.

* P1 contain a signal handler, alarm cycle and array of temperature. Alarm Cycle is used to wake up the program every 1.5 seconds to print out a temperature within the array.

Alarm Cycle

## Results:



# Problem 1.4

This problem involve using 6 processes and a startup program, the startup program fork 6 processes then it save all child PID into an array and writes it into the data module then sleep forever. Startup program is set to sleep because if it exited the other child program will be shutdown.

* Each of the 6 processes are linked to the data module (DM) as shown in the diagram.

***Startup***

***DM***

***P3***

***P2***

***P1***

***P4***

***P5***

***P6***

* Once the children processes are forked, they will be put to sleep for several seconds.
* P1 wakes up and link to DM to retrieve the PID of P2 then send a signal to P2.
* Then P2 wakes up and link to DM to retrieve the PID of P3, P4 and P5 then send signal to each of them. The P5 send signal to P6 and P6 send to P1.

Link to

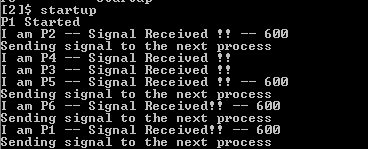
Send to

Fork

Created

* And Process repeats to P1 to P6 and the P6 send it to P1 and the cycle repeats.

## Results:



# Problem 1.5 – Self Driving Car (Smart Car)

***P1***

***P2***

***Startup***

***DM1***

***P7***

***P5***

***P4***

***P3***

Read from

Send to

Fork

Created

Write to

***P7***

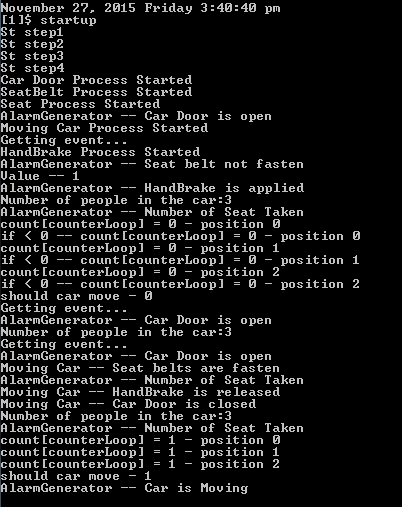
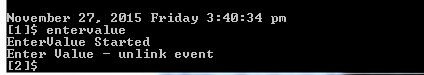
***DM2***

Processes:

* Startup
* P1 – Car Door
* P2 – Seat
* P3 – Seat-belt
* P4 – Hand-brake
* P5 – Generate Alarm
* P6 – CarMove
* P7 – UserEnter to change value
* DM1 – Stores all the PIDs from P1 to P6
* DM2 – Store information allow P1 to P6 to read from, and P7 to write to
* CarInfo.h –Header file contain a list of information.

1. Startup process start the car, and fork all 6 processes from P1 to P6, store each process PID into DM1.
2. P1 check to the DM2 ensure that the car Door is closed. If it’s not close then it will use alarm cycle to wake up every 5.3 second to check if the door is close. If the car door is close then it will send a signal to car move process to tell the process the door is closed.
3. P2 check how many seat is taken in the car according to DM2. It sends signals to Generate alarm process to display there are seat in the car that is taken.
4. P3 check if the seat belt are fasten depending on how many seat(s) is/are taken. If all the seat belts is/ are fasten then it will send a signal to car move process to tell the process the seat belt are fasten.
5. P4 is checking to the DM2 by linking it, to read if the handbrake is applied. If the handbrake is released then it will send signal to car move process to tell the process handbrake is released.
6. P6 – if car move received all signals it will run the car moving check. If all requirement are satisfied then it will send signal to generate alarm process to print out car is moving. When the car is moving this has reach the purpose to this problem 1.5.
7. P5 if any requirement processes is not satisfied, then it will send a signal to this process to generate alarm.
8. P7 It a process that write to the car info memory module (DM2) to change the value in the data module to represent the changing scenarios above.

## Results:



* First Run startup and the process start.
* When all the process the send the messages.
* Then the car send and alert says it’s moving.