

Matlab Profiler

TA 沈煒翔

Code Performance Analysis (Profiling)

- The tool used to analysis code performance : Profiler
 1. Function execution time
 2. Function Execution calls (呼叫了幾次)
 3. CPU usage
 4. Memory usage
 5. Etc ...
- Many IDEs have built-in profiler.
- Matlab also has its own profiling tool.

Usage

```
% Lab4 Problem2 : Detect the R wave in real-time
close all
clear
fclose('all');

profile on % Open profiler and start profiling

while 1
    %
    % Your code here
    %
end

profile viewer % Stop profiling and show profiler result

% close the serial port
fclose(s1);
```

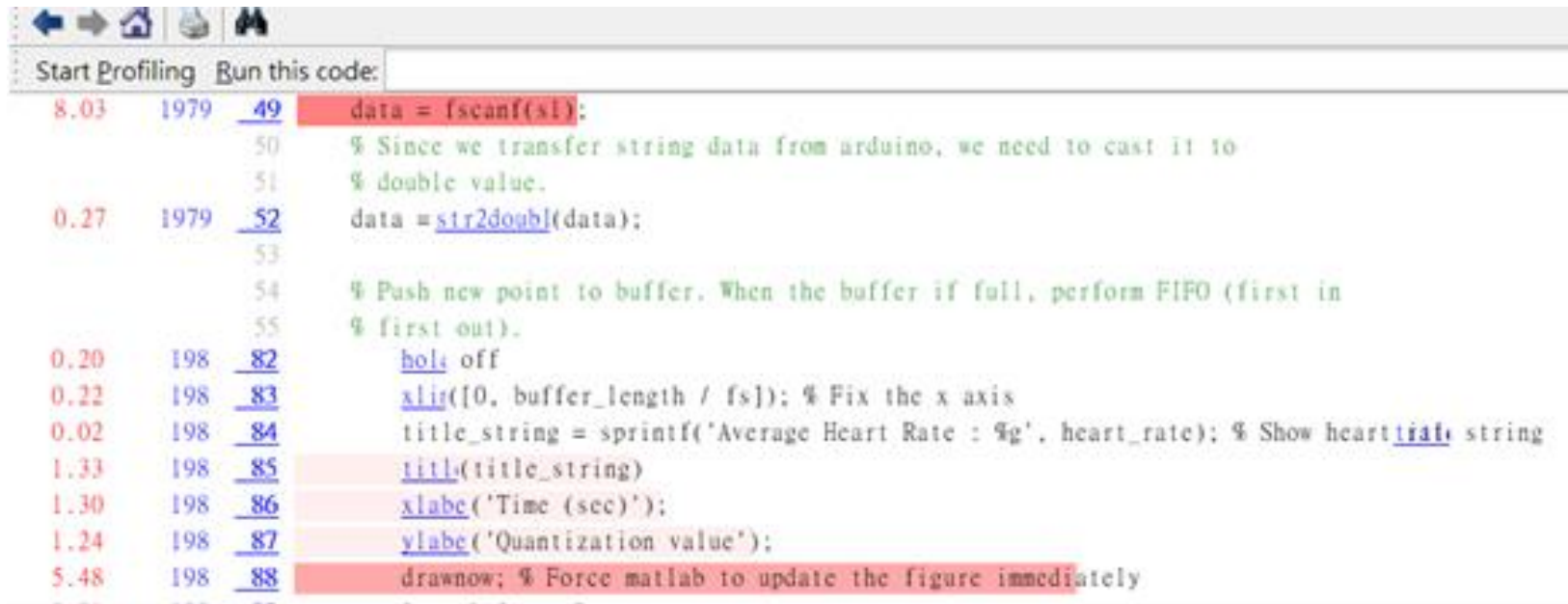
Profiler

File Edit Debug Window Help

Start Profiling Run this code:

Time	Line	Code
	63	% Filter the raw ECG signal
0.09	897 64	filtered_buffer = ECG_filter(data_buffer, mfa_length);
	65	
	66	% Find R-peaks and heart rate
0.10	897 67	[r_peaks, heart_rate] = Find_r_peaks(filtered_buffer, threshold, fs, data_buffer);
	68	
	69	% Show the ECG waveform in the monitor figure
6.26	897 70	plot(t_axis, data_buffer);
0.90	897 71	hold on
1.73	897 72	plot(t_axis, r_peaks, 'o');
0.81	897 73	hold off
0.84	897 74	xlim([0, buffer_length / fs]); % Fix the x axis
0.08	897 75	title_string = sprintf('Average Heart Rate : %g', heart_rate); % Show heart rate string
4.71	897 76	title(title_string)
4.30	897 77	xlabel('Time (sec)');
4.14	897 78	ylabel('Quantization value');
11.67	897 79	drawnow; % Force matlab to update the figure immediately
< 0.01	897 80	end
	81	
< 0.01	899 82	index = index + 1;
	83	% Prevent index from overflowing
< 0.01	899 84	if index >= 50000
	85	index = length(data_buffer);
	86	end
0.09	899 87	end
	88	
< 0.01	1 89	profile viewer

- We can see the most time spent are on “drawnow” and other plotting functions



```
Start Profiling Run this code:
8.03 1979 49 data = fscanf(xl);
50 % Since we transfer string data from arduino, we need to cast it to
51 % double value.
0.27 1979 52 data = str2double(data);
53
54 % Push new point to buffer. When the buffer is full, perform FIFO (first in
55 % first out).
0.20 198 82 hold off
0.22 198 83 xline([0, buffer_length / fs]); % Fix the x axis
0.02 198 84 title_string = sprintf('Average Heart Rate : %g', heart_rate); % Show heart rate string
1.33 198 85 title(title_string)
1.30 198 86 xlabel('Time (sec)');
1.24 198 87 ylabel('Quantization value');
5.48 198 88 drawnow; % Force matlab to update the figure immediately
```

- After some optimization, we can see now the time is bound by transferring data.

```
Elapsed time is 0.051950 seconds.  
Elapsed time is 0.000707 seconds.  
Elapsed time is 0.000535 seconds.  
Elapsed time is 0.000516 seconds.  
Elapsed time is 0.014091 seconds.  
Elapsed time is 0.000823 seconds.  
Elapsed time is 0.006139 seconds.  
Elapsed time is 0.006867 seconds.  
Elapsed time is 0.007832 seconds.  
Elapsed time is 0.010983 seconds.  
Elapsed time is 0.061620 seconds.  
Elapsed time is 0.000810 seconds.  
Elapsed time is 0.000593 seconds.  
Elapsed time is 0.000636 seconds.  
Elapsed time is 0.000936 seconds.  
Elapsed time is 0.000734 seconds.
```

- We can also use tic-toc function to see elapsed time for each loop.
- Average of the elapsed time for 100 loops will be used for Lab 4 evaluation (Demo to your TAs)
- (MATLAB function “tic” “toc”)

Usage

```
% Lab4 Problem2 : Detect the R wave in real-time  
close all  
clear  
fclose('all');
```

```
tic
```

```
while 1  
    %  
    % Your code here  
    %  
end
```

```
toc
```

```
% close the serial port  
fclose(s1);
```

Hint on some optimization

- Filter length
- Sampling rate
- Buffer handling : FIFO means you have to shift the whole buffer for each data acquisition
- Screen update rate : Plotting take much time, so you might not want to update the screen for each data acquisition.