Octave programming

©DuyKy Nguyen, PhD EE @ Unitethc.com 2024-10-22

Octave is an absolute free MatLab with better features
MLab cannot be use as cmdline to run M file but OCTAVE can
I can only use Textpad to run Octave to run right on M file
I found Textpad is the very best editor and I've used in 30 years it since 1994 to run any language C C++ Java Perl Octave Python and bash in cygwin
I've used Tpad not IDE to run those language code without relying on any IDE for them
I can run and edit them at the same time with single key stroke
Whenever I was working with my SW partner I got the result and he was opening IDE
Tpad is cheap only $27 for permant license or yearly license
It support
syntax hilite
run any tool tool
compare file
column mode to fill indexlike to change x[0] below to x[1] x .. x[n] upto the last x[0]
It's ectremely usefull to work with register in Verilog code

x[0]
x[0]
x[0]
x[0]
x[0]
x[0]

I'm morethan happy to help anybody want to use Tpad to run Octave in cmd mode
Just email me at dkn@unitedthc and using LI msg to let me know as your very first likely go to spam folder and I miss it It's just wish to be a nice guy my vgreatest Teacher Buddha has taught me should do for my better better next life the followings;-)
. **Whenever a chance Must Do good thing right away no matter little it is**
. **Never ever do bad thing no matter little it is**
We can carry along those bless as money after death even for the next lifebut not it aelf money They are bless

I also use Ultraedit nut notsupport tool like Tpad but it has a nice feature to search replace compare a block of code not only a sinle word and cout number of some word

1. Function

to be useful function file must be either in a known path either in the same directory or same directory or a path added to Octave in 2 ways
1] create file .octave rc in your user dir with content to add path 2 patheparated by ;
addpath (' C:\Octave\ AI C:\Octave\ XYZ; ', '-begin');
2] or put addpath('.\01' to add dir 01 in the current dir where required function file locates

Func is the very best feature of MLab
It can return single value like
function u=abc(...)
end

It can return multi single values like
function [u v]=abc(...)
end
It support varying func arg in 2 ways

1.1. Using varargin

This is a true varying arg
function z=f2(varargin)
z=0
for k=1:nargin
z+=varargin{k};
endfor
endfunction
function y=f1(a, b, c, d)
y=f2(a, b, c);
endfunction
#{
#}
f1(1, 2, 3)

1.2. Using default arg values

This is not truely vaying arg as it require exactly 2 args but flexible
function z=f1(x=1,y=2)
z=x+y;
endfunction
f1() -->3
f1(2)-->4
f1(4, 5, 6) ERR due to 3 args not 2 as expected

It can return single value like
function u=abc(...)
end
It can return multiple values like
function [uv] =abc(...)
end

1.3. Anymous function

f=@(x, y)=x^2+y^2;
f(2,3)=4+9=13

1.4. Passing function to anotherthru function arg

In 2 ways in easy way usingfunction it self using @ or hardway as obj with feval

Hard way using feval

```
function y=f_a( obj ,x1, x2, x3, x4)
[x1, x2, x3, x4]
y=feval(obj,x1, x2, x3, x4)
endfunction

function z=f_b(a, b, c, d)
z=a +b+ c+ d;
endfunction
k_1=f_b(1, 2, 3, 4)
k_2=f_a('f_b',1, 2, 3, 4)
```

Easy way usingf @

```
function y=f_a( f_b ,x1, x2, x3, x4)
[x1, x2, x3, x4]
y=f_b(x1, x2, x3, x4)
endfunction

function z=f_b(a, b, c, d)
z=a +b+ c+ d;
endfunction
k_1=f_b(1, 2, 3, 4)
k_2=f_a(@f_b,1, 2, 3, 4)
```

2. The similarities

2.1. varying func arg

2.2. default func argvalue

```
function u=abc(x=2, y=3)
u=x+y;
endfunction
abc(

clear all
function y= ff(varargin)
switch nargin

case 1
fprintf('|%d|\n', varargin{1});
case 2
fprintf('|%d, %d|\n', varargin{1}, varargin{2});
case 3
fprintf('|%d, %d, %d|\n',varargin{1}, varargin{2}, varargin{3});
case 4
fprintf('|%d, %d, %d, %d|\n', varargin{1}, varargin{2}, varargin{3}, varargin{4});
endswitch
fprintf('$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n');
endfunction
```

ff(1);

ff(1, 2);
ff(1, 2, 3);
ff(1, 2, 3, 4);

Semicolon is optional at the end to display or not var values
Index using () start from 1 not 0
array using spquare bracket [. . .] vector [1 2 3] with 3 item
matrix [1 2 3; 4 5 6] with 2 rows 3 colsusing ; or new line as
[1 2 3
4 5 6]
function name abc must be in a file abc.m A trick to get around this is to put clear all at top of a file to use fuction in any mcode

IF requires no parens()for single condition like
if x <= 1
Unless compound cond like
if (x<=5 && y>=10)
statement code blk starts from statement to another or end
switch n
case 1 start case 1
.
.
.
case 2 end case 1 new case 2
.
.
.
otherwise end case 2
end end switch statement

if x >= 1 start if
.
.
.
elsif x <=10 end if, start elsif
.
.
.
end end if statement

Varying function arg
clear all
function y= ff(varargin)
switch nargin number of input arg function

case 1
fprintf('|%d|\n', varargin{1}); first arg
case 2
switch nargin number of input arg function

case 1 start case 1
fprintf('|%d|\n', varargin{1}); first arg
case 2 end case 1 start case 2
fprintf('|%d, %d|\n', varargin{1}, varargin{2});
case 3 end case 2 start case 3
fprintf('|%d, %d, %d|\n',varargin{1}, varargin{2}, varargin{3});
case 4 end case 3 start case 4
fprintf('|%d, %d, %d, %d|\n', varargin{1}, varargin{2}, varargin{3}, varargin{4});
endswitch end switch
fprintf('$$\n');
endfunction
ff(1);

ff(1, 2); 2 input arg
ff(1, 2, 3); 3 input arg
ff(1, 2, 3, 4);

3. Differences

MLab use only IDE to run the code, no option to run code with cmdline
We can use cmd-line Octave-cli to run mcode using textpad and running octave-cli at its folder
uniary operator ar possible in Octave but not MLaAb
MLab use % for comment confuusing with remainder op 3%2=1 Octave use #
Blk comment use {} with coment #{ open blk comment #} to close
x++
x+=n
Octave use either cmdline or IDE to run
MatLab uses end for all function if for
Octave uses endif endfor end function end swich endfunction
end also use for last index
Varying function arg
clear all
function y= ff(varargin)
switch nargin number of input arg function

case 1
fprintf('|%d|\n', varargin{1}); first arg
case 2
fprintf('|%d, %d|\n', varargin{1}, varargin{2});
case 3
fprintf('|%d, %d, %d|\n',varargin{1}, varargin{2}, varargin{3});
case 4
fprintf('|%d, %d, %d, %d|\n', varargin{1}, varargin{2}, varargin{3}, varargin{4});
endswitch
fprintf('$$\n');
endfunction
ff(1);

ff(1, 2); 2 input arg
ff(1, 2, 3); 3 input arg
ff(1, 2, 3, 4);

4. Plotting

It's alot easier to use legend in Octave than MLab
Belows isx axexample

```
clear all
x= 0:1e-2: 2*pi;
 y_s01 = sin (x);
 y_s02 = cos (x );

clf
clf(gcf)


plot(x, y_s01, 'r*',x, y_s02, 'b*'), grid; # grid for grid


title('SS') #title must be below plot ie naming after plotting
# set plot propeties like

set(gca, 'linewidth', 6, 'fontsize', 22, 'fontweight', 'bold' )
# legend just put all plot in order, it auto to display properly in order

  h = legend ('Sin', 'Cos','location', 'NorthEastOutside'); #Follow order in plot
# NorthEastInside is default location , i.e if not using 'location' at all
 set (h, 'fontsize', 20, 'fontweight', 'bold');
 pause( 22 )
```