j’ai enfin réglé le problème de require :

On marque les fonctions telle que :

myApp.clicked()

Puis on passe la commande suivante :

browserify demo.bs.js --standalone MyApp >bundle.js

L’efficacité et le temps d’exécution :

Afin de pouvoir mieux comparer le temps d’exaction des trois méthodes on va comparer ce temps là en deux phases. Premierement on compare le temps moyen de compilation des fichier .ml lui même divisé en trois temps système, utilisateur et réel puis dans un deuxième temps on comparera le temps d’exécution des fichiers .js .

La partie Ocaml :

J’ai fait un script shell ainsi qu’un programme caml qui lance la commande time sur un programme caml (un programme qui affiche une simple fenêtre alert avec un message) pour les 3 méthodes et pour un nombre d’exécution passé en paramètre , ce script renvoie donc le temps moyens d’exécution réel , système et utilisateur pour les trois méthodes sur un nombre x d’exécution passé en paramètre.

La partie JavaScript :

Comment on mesure l’efficacité en temps en JavaScript :

Cette efficacité n’est mesuré que par le temps réel.Ainsi on choisi un point de début au début du script et un point à la fin du script puis on affiche le résultat de soustraction du deuxième point par le premier point pour se faire il y a plusieurs méthodes en JavaScript :

La meilleure méthode de mesure de l’efficacité d’un code en JavaScript jusqu’à présent est la fonction performance :

ainsi on pourra mesurer l’efficacité du code suivant en milliseconde :

var t0 = performance.now() ;

doSomething() ;

var t1 = performance.now() ;

console.log(t1-t2 + millisecondes )

Js\_of\_ocaml :

La méthode performance n’est pas encore implémentée en js\_of\_ocaml on peut toujours passer pour la methode un peu plus ancienne et moins efficace avec les objets date :

var t0 = date.now() ;

doSomething() ;

var t1 = date.now() ;

console.log(t1-t2 + millisecondes )

var t0 = performance.now() ;

doSomething() ;

var t1 = performance.now() ;

console.log(t1-t2 + millisecondes )

puis on passe aux methodes

Pb rencontr

performance.now