070711 13h30 franck Remarques générales

- il faut ajouter les sources dans le svn
- Et il faut un script pour générer une arborescence pour la compilation
    - ♦ ici on va avoir différentes personnes qui vont prendre les sources :
        - ◊ nous pour faire vivre le système
        - ◊ les etudiants pour lire le code et le modifier Je ne sais pas s'il faut la même arbo
- Pour les docs des Pthreads, il faut dire que la docs est celle officielle, donc avec les noms des auteurs
- Pour la lecture du code, il vaut mieux partir d'une application, que l'on désassemble

070711 15h10 franck Lecture du code

- a quoi sert le répertoire mk dans src
- dans les sauvegarde des registres, on pourrait leur donner leur nom symbolique, ce serait peut etre plus parlant
- il va falloir relire tous les commentaires pour corriger l'anglais.
- pour le dead end, il faut quand meme mettre du code qui bloque le processeur apres affichage d'un petit message.
- il faut regarder le comportement du cache lors de la sauvegarde
- dans reset.s, je ne suis pas sûr que les directive glob ent et end soient utiles, a vérifier
- Est-ce qu'on remet à zéro les zones bss
- a quoi sert .extern dans services
- en fait je pense que dans tous les codes assembleur, il faut donner les noms symboliques des registres soit dans l'instruction soit dans le commentaire
- pourquoi avoir écrit xor $2, $2, $2 et pas addi $2, $0, 0
- est-ce que les double underscore sont une convention "officielle"
- pourquoi y a t il un main dans le phtread ?
- Pour l'indentation du code : indent -kr -bli0 -i4 -ts0 -l128 -npsl fichier.c donne un code assez compact
- bien la macro PANIC mais il en faudrait une qui ne fasse rien en mode DEBUG
- il faut un mode DEBUG où on a des affichages
- il faut des compteurs pour le profiling (IMPORTANT), c'est un profiling soft parceque je ne veux pas qu'on change la plateforme. le pb du soft est que l'on vient perturber le fonctionnement normal, donc il faut quantifier la perte pour éventuellement rectifier les résultats.