Rapport td2
Systeme d’exploitation

I Bilan

Cette fois les programmes utilisateurs peuvent être lancés, `do_ThreadCreate(int f,int arg)` se lance bien, mais aucun thread ne se lance. En effet après l’appel à `nt->Start(StartUserThread, (void*) ptr );` (avec nt étant le nouveau thread), ne semble pas s’effectuer correctement voir pas du tout, car Il ne semble jamais passer dans `static void StartUserThread(void*schmurtz)` (voir partie test). Pourtant `do_ThreadCreate` continue bien ses instructions.

III Limitations

Ce qui limite ce projet est la non-création de thread et le fait de pas « désallouer » l’espace mémoire du thread (car jamais lancé et donc jamais testé). Toute la seconde partie sur la gestions de plusieurs thread n’a pas pu être implémentée.

IV Tests

Les test ont été effectué avec makethread.c, qui utilise l’appel système de de création de thread. Aucune trace de la première fonction que la thread doit exécuter n’a été détecté, les tests avec les debug, montre que pourtant `nt->Start(StartUserThread, (void*) ptr );` envoie bien deux adresses correspondant aux paramètre de `do_ThreadCreate(int f,int arg)` , de plus avec le memory.svg il ne semble pas non avoir de zone mémoire utilisée pour le thread, et lancer avec l’option -s, montre que l’appel système du main makethread.c sont bien pris en compte, mais pas celui de la fonction que le thread doit appeler. On peut lire aussi avec cette option que le PC n’a pas l’air de changé de valeur après l’appel système (autre changement que l’incrémentation vers la prochaine instruction).