

Practica 3 – Sistemes Operatius II

- **Comentaris de la Implementació:**

En aquesta tercera practica ens centrem en la implementació de un menú textual que deixa escollir que funcions executarà el programa en cada moment. El menú ha de contindre 5 opcions: La primera, referida a la funcionalitat implementada a les practiques anteriors, es la creació d'un arbre a memòria. La segona i la tercera, es demanen que les implementem nosaltres, son funcions que guarden (2) i carreguen (3) les dades del arbre binari a un fitxer a disc (no amb el mateix format que el fitxer .csv). La quarta funció demana que ens comuniquem mitjançant una *pipe* amb el *gnuplot* per a passar-li el retràs mig per dies entre dos aeroports i fer una gràfica amb aquestes dades. La ultima funció correspon a sortir del programa (5).

- **Menú de texte i creació del arbre:**

El primer que es demanava era la implementacio de un menú de texte per a poder interactuar amb les diferents funcions del programa aixi com agafar tota la funcionalitat de les practiques anteriors i preparar-la dins de la funció (1) del menú. Aquest punt no te gaire secret, ja que el menú de texte es dedica a imprimir les opcions per a informar a l'usuari, capturar un enter desde teclat i finalment decidir el fluxe de execucio amb un bloc *switch-case*. En aquesta part del codi '*main.c*' farem tot el control de errors i el control de memoria (que l'arbre estigui en memoria per a salvar-lo en disc, etc.).

- **Guardat i carregat de les dades del arbre a disc:**

Per a les funcions de guardar i carregar vem haver de fer diverses iteracions al codi fins que vem aconseguir tenir-ho funcionant correctament. Ambdues les hem situat al C del arbre, ja que son funcions que es podrien cridar des de un altre programa diferent que utilitzes també la estructura arbre binari. El primer va ser fer la funció de guardat, vem decidir que abans de cada llista guardaríem la seva longitud, així a la hora de llegir podríem saber quants elements s'han de iterar. Tant per guardar com per llegir ho fem de manera recursiva, per poder recórrer el arbre per 'dreta' i 'esquerra'.

El ordre en que es guarden es relativament lliure mentre la funció de llegir l'arbre de disc utilitzi el mateix, nosaltres hem decidit el ordre logic de 'key, longitud de la llista, key llista i elements de la llista per cada dia.

Per a carregar utilitzem la instrucció de C feof(file) per saber quan s'ha acabat el fitxer i el us de 'memcpy' per a copiar el contingut de la key correctament, ja que aquí vem tenir el problema de que obteníem la referencia només.

- **Comunicacio amb el procés fill 'gnuplot':**

Per a la comunicació amb 'gnuplot' vem decidir que la inicialització de la pipe i el tancament els faríem al main.c i vem crear una funció independent que amb la pipe inicialitzada ja fa la crida al gnuplot i ens mostra la gràfica. A comentar d'aquet punt podríem dir que vem haver de trobar una manera de que el 'gnuplot' ens mantingues la gràfica en pantalla, després d'una breu cerca vem trobar que era la comanda '-persistent' que únicament s'havia de afegir a on cridàvem al 'gnuplot' quan fèiem la pipe. Vem decidir utilitzar fflush() al final de enviar les dades per forçar que aquestes fossin escrites a la pipe tot i no tenir el buffer de usuari ple.

Un problema que hem trobat i que no vem solventar del tot va ser que al demanar al usuari els dos aeroports per a fer la gràfica, el valgrind ens diu que hi ha un salt condicional que depèn de una variable no inicialitzada, i es de fet així, però perquè la variable que es refereix es la que utilitzem al 'scanf' per rebre del teclat els aeroports.

Teniem un while la longitud de les dos entrades no fos 3 es mantingués demanant la entrada, però ens donava uns cuans errors m's degut a que al malloc d'aquestes variables no reservàvem més memòria que la necessaria per a 3 caracters, no tindria sentit augmentarla per si l'usuari decidís entrar 6 caracters, així que vem suprimir aquesta part del codi i vem deixar-ho en un 'if' i suposant que el usuari ho introdueix correctament.