



Tutoriel 5 - Structures, fichiers d'entête et analyse de données.

Bon retour à la caverne du C. Ce tutoriel donne un exemple de programme construit à partir de plusieurs fichiers sources C et compilé avec un Makefile. Avant de continuer, comment vous en êtes-vous sortis avec le défi précédent ?

Solution du défi

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/sysinfo.h>
int main() {
    int i = 0;
    float ramUsed;
    char gpCmdOne[250], gpCmdTwo[250], systemCmd[1000];
    FILE *cmdPtr = 0, *outPtr = 0;
    char c, fileName[100], *strPtr = 0;
    struct sysinfo info; /* Structure sysinfo pour stocker l'état. */
    cmdPtr = popen("hostname","r"); /* Lance la commande hostname. */
    if(!cmdPtr) return 1;
    strPtr = &fileName[0]; /* Pointeur sur la chaîne. */
    while((c=fgetc(cmdPtr)) != EOF) { /* Accès à chaque caractère. */
        *strPtr = c; /* Fixe la valeur du caractère. */
        strPtr++; /* Va à la position suivante dans le tableau. */
    }
    pclose(cmdPtr); /* Ferme le fichier hostname. */
    strPtr--; /* Va un élément en arrière pour ignorer la fin de ligne. */
    sprintf(strPtr,"-data.txt"); /* Ajoute le suffixe. */
    printf("%s\n",fileName);
    outPtr = fopen(fileName,"w"); /* Ouvre le fichier de sortie. */
    if(!outPtr) return 1; /* Retourne une erreur s'il est impossible d'ouvrir
le fichier */
    for(i=0;i<60;i++) {
        sysinfo(&info); /* Lit les infos système */
        ramUsed = info.totalram - info.freeram;
        ramUsed /= 10240.0;
        fprintf(outPtr,"%d %f %d\n", i, ramUsed, info.loads[0]); /* Écrit la ram
utilisée. */
        usleep(500000); /* Pause d'1/2 seconde. */
    }
    fclose(outPtr); /* Ferme le fichier de sortie. */
    /* Maintenant, affichage des données */
    sprintf(gpCmdOne, "plot \'%s\' using 1:2 title \'%s\' ", fileName, "Ram
utilisée");
    sprintf(gpCmdTwo, ", \'%s\' using 1:3 title \'%s\'\n", fileName, "Charge");
    /* Crée la commande complète, avec le tube vers gnuplot */
    sprintf(systemCmd, "echo \'%s\' | gnuplot -persist", gpCmdOne, gpCmdTwo);
    system(systemCmd); /* Exécute la commande système. */
    return 0; /* Retour réussi vers le système. */
}
```

La solution met en œuvre des fonctions et des techniques vues précédemment. Des façons plus simples existent pour créer le nom de fichier à partir de l'hôte. Le C propose un entête string.h avec des fonctions