R101 : Initiation au développement

TD 13 : la programmation modulaire en C

… en vous appuyant sur votre cours sur la programmation modulaire :

Exercice 1 : l'instruction #include

Vérifiez, à l'aide de l'exemple ci dessous que l'instruction du préprocesseur #include effectue bien une copie du fichier auquel elle fait référence.

fonctions.h

```
int fonction(int a);
void procedure(float x, float y);
```

main_exo1.c

```
# include "fonctions.h"

int main(){

        return 0;
}
```

1. Quelle instruction permet de visualiser le fichier créé par le pré-processeur ?
2. Générer et visualiser le fichier créé par le pré-processeur.

Exercice 2 : Découpez votre programme du TD12 « File statique en langage C » en utilisant les règles présentées en cours.

Pour rappel, votre application devra comporter les fichiers suivants :

Nom du fichier	Rôle
const.h	déclaration des constantes symboliques
types.h	déclaration des types utilisateur
globales.h	déclaration des constantes
globales.c	définition des constantes
fonctions.h	déclaration des prototypes des fonctions
fonction.c	définitions des fonctions
main_file.c	définition du programme principal

1. Vous effectuerez la compilation de chaque fichier source (au fur et à mesure) afin de générer les différents fichiers objets.

   Quelle option de gcc permet de faire une compilation séparée ?

   Quelle est l'extension des fichiers objet ?

2. Vous effectuerez ensuite l'édition de lien qui vous permettra de générer le fichier exécutable.
3. On vous demande maintenant d'écrire un fichier système qui va exécuter l'ensemble des instructions préalables et lancer l'application.

Exercice 3 : complétez les messages de la file avec la date et l'heure d'émission :

1. Présentation :

La fonction time() du langage C permet d'obtenir le temps écoulé depuis le premier janvier 1970 à 00:00:00, sous forme d'un entier positif.

Voici un programme simple qui permet de connaître cette valeur :

```
# include <stdio.h>
# include <time.h>
int main(){
        long temps;
        temps = time(NULL);
        printf(" nombre de secondes depuis le 1 janvier 1970 :%d\n", temps);
        return 0 ;
}
```

2. Vous disposez des deux fichiers suivants :

- fonctions_date.h : déclaration des prototypes des fonctions de gestions de la date.
- fonctions_date.o : résultat de la compilation séparées du fichier fonctions_date.c dans lequel deux fonctions ont été implémentées :
  - le fonction date2str() génère la date sous la forme d'une chaîne de caractères à partir du temps (en secondes) écoulé depuis le 1 janvier 1970.
  - la fonction date2int() retourne le temps (en secondes) à partir d'une date donnée sous la forme d'une chaîne de caractères.

Remarque : le format des chaînes de caractères devra impérativement être le suivant :

```
jj/mm/aaaa_hh:mm:ss
```

jj : numéro du jour du mois
mm : numéro du mois de l'année
aaaa : année
hh : heures
mm : minutes
ss : secondes

3. Vous pouvez tester la manipulation des dates avec le programme suivant :

```
#include <stdio.h>
#include <time.h>
# include "fonctions_date.h"

int main( int argc, char * argv[] ) {
   char date[MAX_SIZE];
   long temps;
   temps = time(NULL); // date et heure de l'instant
   printf("%ld\n",temps);
   date2str(temps, date);
   printf("%s\n", date);

   temps = date2int(date);
   printf("%d\n", temps);
   return 0;
}
```

4. Modifiez votre application "gestion d'une file" en ajoutant le champ "date" (au format chaîne de caractères) dans la structure t_element.

5. Complétez votre application en ajoutant la procédure supprime_anciens_date() qui supprime les messages antérieurs à une date donnée.