



Programmer une horloge en langage C

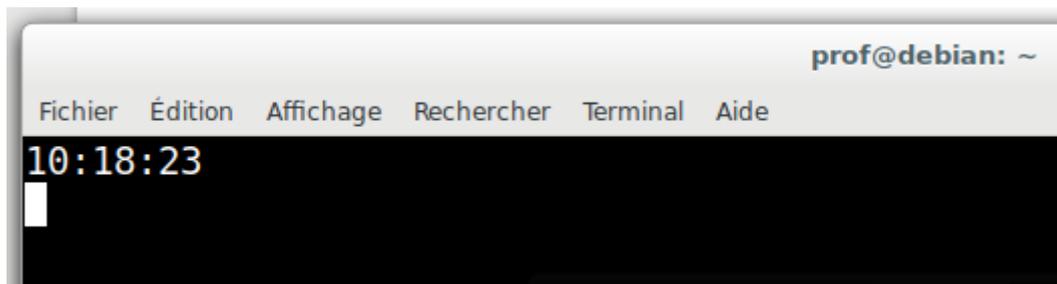
Nom : Prénom : Classe : Date :	Appréciation : <i>Remarque : n'utilisez pas la couleur rouge qui est réservée au correcteur</i>	Note : /20
---	---	---------------------------------

tâches métier : créer un programme compilé en C	durée : variable
--	-------------------------

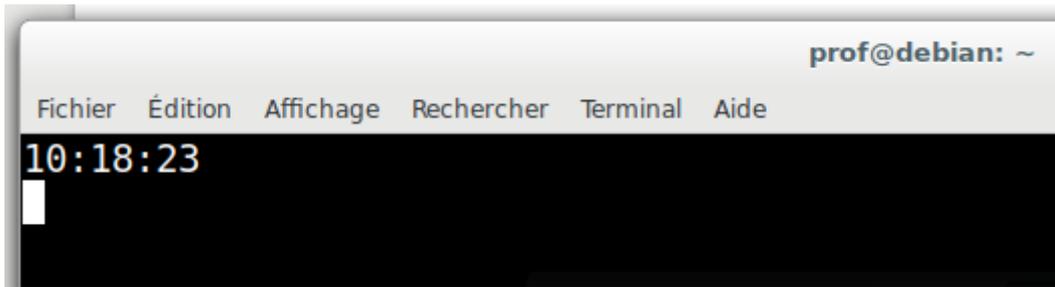
Matériel :

Compétences et savoirs principalement visées : voir sujet

Objectif : écrire un programme en C qui affiche une horloge avec les heures, minutes et secondes



Objectif : écrire un programme en C qui affiche une **horloge** avec les heures, minutes et secondes :



1ère étape : Écrire la structure du fichier source en C

- Le fichier du code source présente la structure suivante ; crée ce fichier avec la commande « nano heure.c » :

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>

/////////////////////////////////////////////////////////////////
// heure
//
// time() : La valeur renvoyée correspond au nombre de secondes écoulées depuis le premier janvier 1970.
//
// Syntaxe : heure
//
/////////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////// Main ///////////////////////////////////////////////////////////////////

int main (void)
{
    int i;
    time_t secondes;
    struct tm instant;

    time(&secondes);
    instant=*localtime(&secondes);
    printf("%d/%d ; %d:%d:%d\n", instant.tm_mday+1, instant.tm_mon+1, instant.tm_hour, instant.tm_min, instant.tm_sec);

return 0;
}
```

Ici on doit déclarer
Les bibliothèques
utilisées

Les lignes commençant
Par les caractères //
Sont des commentaires

Le programme principal commence par
int main (void) {
et se termine par **return 0;}**

Les déclarations de variables

- Tapes « CTRL-O » puis « Entrée » puis « CTRL-X » pour enregistrer et sortir du programme nano

2ème étape : compiler le code source en C

- Dans le fichier **heure.c**, créé à la page précédente, il y a plusieurs lignes de codes (celles qui sont en **noir**), récupérées sur internet, qui sont censées renvoyer la date et l'heure ; mais avant de pouvoir les tester, il faut **compiler** le programme avec la commande suivante :

```
gcc -Wall heure.c -o heure
```

- Si il y a des messages d'erreur, corrige le fichier `heure.c`, puis essaie à nouveau de le compiler, jusqu'à ce qu'il n'y ai plus de messages d'erreur
Remarque : les « Warning » ne sont pas des erreurs, juste des avertissements.
- Lance le programme « heure » compilé avec la commande suivante :

```
./heure
```

Bilan à la fin de la première étape

Nous sommes capable, grâce à la commande trouvée, d'afficher la date et l'heure sous cette forme :

```
cv@W7E107-PROF:~/Documents/dev/C$ ./heure
23/3 ; 16:3:41
cv@W7E107-PROF:~/Documents/dev/C$ █
```

3ème étape : formater la chaîne récupérée pour ne garder que les heures, minutes et secondes

Le problème avec la commande utilisée, c'est qu'en plus de retourner l'heure, elle retourne aussi la date, dont nous ne voulons pas : il faut donc trouver une méthode pour ne garder que la partie qui nous intéresse.

```
23/3 ; 16:3:41
```

Comment faire ?

- Repère dans le code source, la ligne qui affiche cette chaîne de caractère (en anglais, « afficher » se traduit par « print ») ; entoure-la en rouge.

int main (void)

```
{
    int i;
    time_t secondes;
    struct tm instant;

    time(&secondes);
    instant=*localtime(&secondes);
    printf("%d/%d ; %d:%d:%d\n", instant.tm_mday+1, instant.tm_mon+1, instant.tm_hour, instant.tm_min, instant.tm_sec);

return 0;
}
```

- Modifie cette ligne pour obtenir le résultat souhaité ; tu peux faire autant de tentative que nécessaire ; *pour réussir, tu dois prendre le temps de bien comprendre le fonctionnement de la commande « printf »*
- Recopie ci-dessous la ligne modifiée :

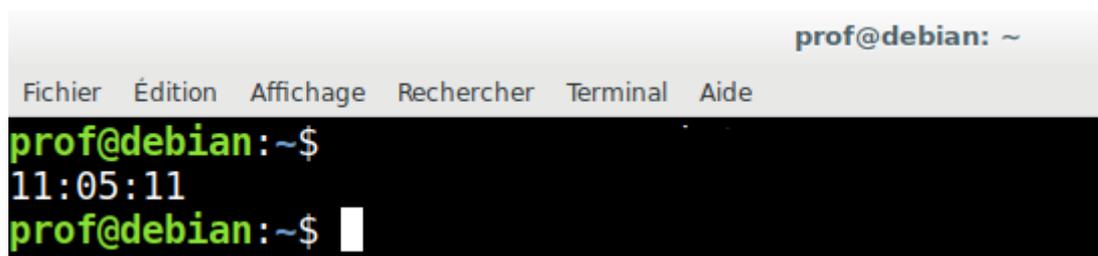
Information sur : la commande « printf »

La commande **printf** possède une syntaxe bien particulière ; il faut du temps pour la comprendre ; c'est la principale difficulté ici ; dans les paramètres de **printf**, il y a 2 parties bien distinctes : d'abord, une « chaîne de formatage », puis une liste des paramètres qui vont remplir cette chaîne ; *fais des recherches sur internet, pour comprendre leurs rôles respectifs.*

Dans l'exemple ci-dessus, le premier **%d** de la chaîne de formatage sera remplacé par la valeur de la variable **instant.tm_mday+1**; le second **%d** de la chaîne de formatage sera remplacé par la valeur de la variable **instant.tm_mon+1**; etc, etc, ...

Bilan à la fin de la troisième étape

Nous sommes capable, grâce à la commande trouvée, d'afficher l'heure, sans la date, sous cette forme :



```
prof@debian: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
prof@debian:~$
11:05:11
prof@debian:~$
```

4ème étape : Rafraîchir automatiquement l'affichage de l'heure toutes les secondes

- Modifie et complète le fichier `heure.c` ; tu dois remplacer le rectangle jaune par une instruction C qui permet de faire une pause de 1 seconde :

```
int main (void)
{
    time_t secondes;
    struct tm instant;

    while (1==1) {
        time(&secondes);
        instant=*localtime(&secondes);
        printf("%d:%d:%d\n", instant.tm_hour, instant.tm_min, instant.tm_sec);
        
    }
```

While (1=1) { est le début de la boucle

} est la fin de la boucle

- Tapes « CTRL-O » puis « Entrée » puis « CTRL-X » pour enregistrer et sortir du programme nano
- Compile le programme
- Lance le programme « heure » avec la commande :

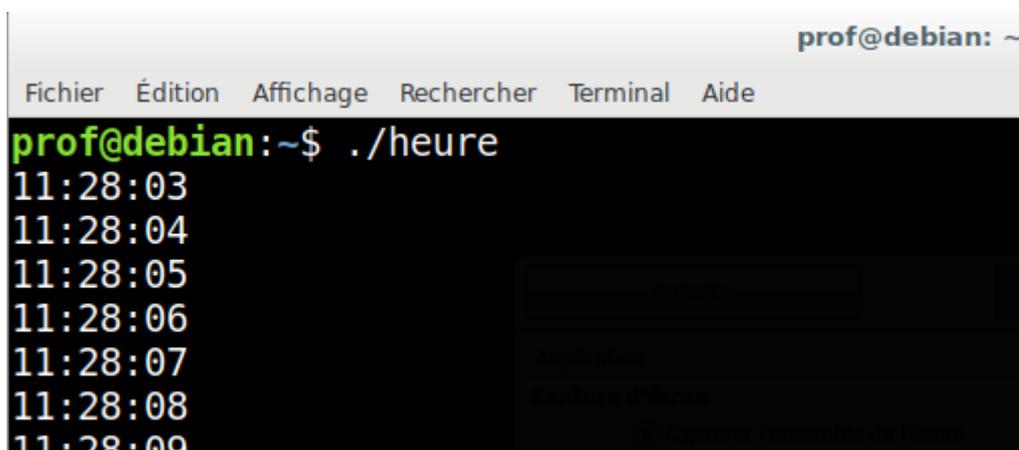
```
./heure
```

Bilan à la fin de la quatrième étape

Nous obtenons l'affichage automatique de l'heure toutes les 1 secondes par la commande :

```
./heure
```

L'affichage ressemble à :



```
prof@debian: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
prof@debian:~$ ./heure
11:28:03
11:28:04
11:28:05
11:28:06
11:28:07
11:28:08
11:28:09
```

5ème étape : Effacer automatiquement l'écran avant d'afficher la nouvelle heure

L'inconvénient de notre programme en l'état actuel, c'est qu'il affiche les heures successives les unes après les autres, ce qui n'est pas très lisible ; **il faudrait effacer l'heure précédente avant d'afficher la nouvelle heure.**

Cherches sur internet une ligne de programme C qui permet d'effacer l'écran du « Terminal ». **indice** : c'est de la forme : « **system(quelquechose);** »

- Complète le fichier heure.c, en rajoutant tout ce qui manque :

```
int main (void)
{
    time_t secondes;
    struct tm instant;

    while (1==1) {
        time(&secondes);
        instant=*localtime(&secondes);
        printf("%d:%d:%d\n", instant.tm_hour, instant.tm_min, instant.tm_sec);
    }
}
```

- Tapes « CTRL-0 » puis « Entrée » puis « CTRL-X » pour enregistrer et sortir du programme nano
- Compile le programme
- Lance le programme « heure » avec la commande :

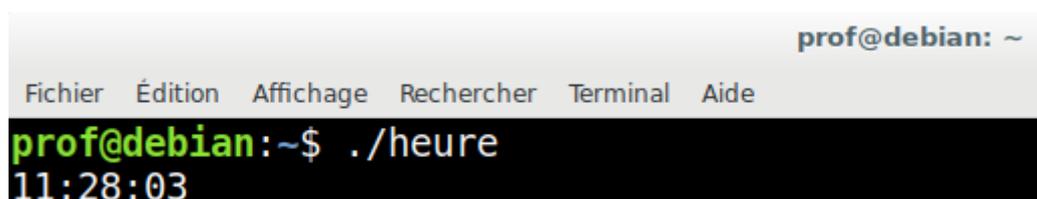
```
./heure
```

Bilan à la fin de la dernière étape

Nous obtenons l'affichage automatique de l'heure toutes les 1 secondes par la commande :

```
./heure
```

L'affichage ressemble à (on voit les secondes défiler):



```
prof@debian: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
prof@debian:~$ ./heure
11:28:03
```