esiea
INGÉNIEUR·E·S D'UN NUMÉRIQUE UTILE

C-SYSTÈME

Fiche : Débogueur GDB/DDD

Table des matières

Les prérequis au débogage.....2
Utilisation simple.....2
Utilisation avancée.....3
Debogage dans une fonction.....3

Les prérequis au débogage

Le débogueur du projet GNU s'appelle « gdb ». Il possède une interface graphique « ddd » (Data Display Debugger ». Ce debogueur est présent dans toutes les distributions Linux. Il s'installe donc au moyen des gestionnaires de paquetages traditionnel (urpmi, apt, etc.). Pour être exploité, le débogueur doit trouver dans le programme à analyser plusieurs informations complémentaires lui permettant de proposer un environnement convivial

By Rexy

(debugging symbols). À titre d'exemple, il doit trouver le nom et l'emplacement du fichier source afin de l'afficher dans son interface. Il doit trouver le nom de vos variables afin de pouvoir les suivre.
L'option « -g » indique que « gcc » doit ajouter ces informations au binaire. Cette option ne doit être exploitée que pendant la phase de conception d'un programme.

```
gcc -g -o nom_du_binaire  nom_du_fichier_source.c
```

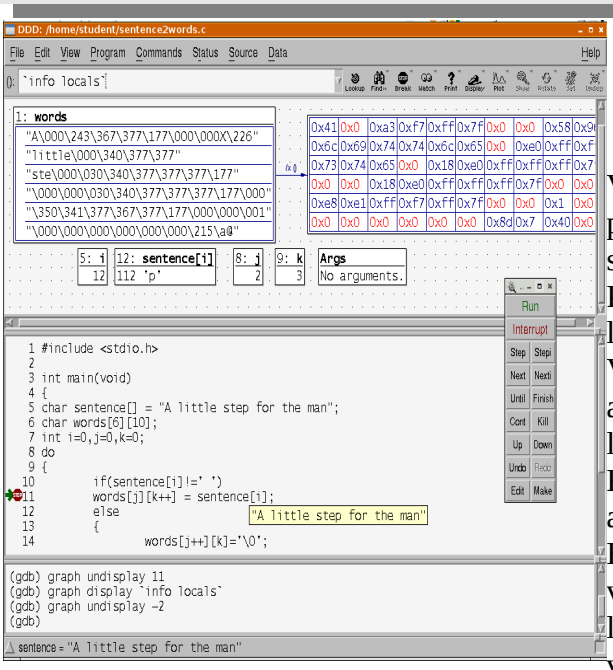
Utilisation simple

Considérant le programme suivant qui décompose une phrase en un tableau de mot (sentence2words.c).

```
#include <stdio.h>

int main(void)
{
char sentence[] = "a little step for the man";
char words[6][10];
int i=0,j=0,k=0;
do
{
	if(sentence[i]!=' ')
		words[j][k++] = sentence[i];
	else
	{
		words[j++][k]='\0';
		k=0;
	}
} while(sentence[i++]!='\0');
for(i=0;i<6;i++)
	printf("%s\n",words[i]);
return 0;
}
```

```
gcc -g -o sentece2words sentence2words.c
ddd sentence2word
```



- le cadre d'affichage du code source ;
- le cadre inférieur est une console de commandes « gdb » ;
- la fenêtre volante de commandes (command tools) permet de contrôler l'exécution du programme.

Vous pouvez ajouter la fenêtre de résultat d'exécution du programme (View + « execution window ») ainsi qu'un cadre de suivi des variables (View + « data window »).
Pour numéroter les lignes du code source : (Source + « Display line numbers »).
Vous pouvez positionnez des points d'arrêt via un « click droit » au niveau des numéros lignes du code source, puis lancer l'exécution.
En glissant la souris sur les variables, une fenêtre contextuelle affiche leur valeur actuelle.
En double cliquant sur une variable (ou en la sélectionnant), vous l'ajouter au cadre de suivi (cadre supérieur). Dans cette liste, chaque variable peut être affichée sous différentes formes via le menu contextuel (cf. tableau « words » affiché en décimal et en hexadécimal). Au passage, on remarque que les entrées non initialisées contiennent l'état de la mémoire.
Vous pouvez afficher toutes les variables de votre programme en une seule opération via le menu (Datas + « Display local variables »).
Dans la fenêtre volante de commandes : « Next » permet d'avancer pas à pas, « Cont » poursuit jusqu'au prochain point d'arrêt.
Si votre programme attend des arguments, « ddd » vous permet de les entrer via le menu « program » « run » (ou <F2>). !!! le bouton « run » du panneau de contrôle n'offre pas cette fonctionnalité.

Utilisation avancée

Débogage dans une fonction

TODO - option1 : utilisation de DDD sur un programme exploitant des fonctions (affichage de la pile des fonctions (« stack ») / des variables locales et globales). Utiliser pour cela votre programme « sentence2word » avec une fonction (cf. §4 du LAB0).

Débogage d'un fichier 'core '

TODO - option1 : Qu'est ce qu'un fichier « core » ? Comment est-il généré ? Faite un programme simple avec fonction qui en génère un.

TODO – option 2 : réaliser cette fiche pour le débogueur de l'IDE « codeblock »