

LAB1

Prise en main de l'environnement



Table des matières

1 Objectifs.....	2
2 Rappels et mise en place de l'environnement.....	2
3 Travaux préparatoires	2
4 Connaissance système	2
5 Source, code assembleur, binaire et débogage.	3
6 Évolution de code	4
7 Amélioration de la fiche « débogage »	5

By Remy



2 Objectifs

Ce LAB permet de mettre prendre en main l'environnement minimal pour aborder les notions de programmation système.

- Documents :
 - Supports : « Linux – Les commandes » et « Linux – L'éditeur Vi » ;
 - Fiches : « Compilation » et « Débogage ».
- Notions abordées : l'environnement de développement, le débogage.
- Commandes et fichiers exploités : gcc, commandes de gestion de fichiers, ddd.
- Travail à rendre : Vous devrez répondre directement à plusieurs questions au sein de ce document. Vous le copierez sur Moodle sous le nom : « LAB0_noms.odt ».

3 Rappels et mise en place de l'environnement

Les principes abordés dans ce document sont conformes aux préconisations « Standard Unix Specification V4 (SUSV4) ». Les distributions Linux, à travers la standardisation « Linux Standard Base (LSB) » sont conformes à « SUSV4 ».

Pour développer, vous devez utiliser les outils standards GNU (gcc + gdb/ddd) et un éditeur de texte type Vi/Emacs/Gedit, l'utilisation d'un IDE est interdite.

4 Travaux préparatoires

Pour vous familiariser à nouveau avec l'environnement :

- Lisez et effectuez les commandes décrites aux chapitres §1 à §3.7 du support « Linux - Les commandes.pdf » ;
- Lisez les rubriques « topic1 » et « topic2 » du cours LPIC1V3 (Linux Professional Institute Certification) disponible sur Moodle (dans « accueil » + « espace commun » + « Certification Linux »). Effectuez plusieurs auto-test de ces deux « topics ».

5 Connaissance système

Objectif : Connaître le système sur lequel vous êtes en train de travailler : nom de la distribution, version du noyau, capacité de la machine (CPU, mémoire, carte mère et bios, cartes additionnelles), topologie du disque (marque, taille, partitionnement) ?

- Copiez ci-dessous les informations recherchées (les commandes utilisées et le résultat de celles-ci pour votre système).

CPU : intel(R) Core™ i7-3667u CPU 2.00ghZ

```
srikanth@srikanth-VirtualBox:~$ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 58
model name     : Intel(R) Core(TM) i7-3667U CPU @ 2.00GHz
```

Taille mémoire vive : 4430768 kb

```
srikanth@srikanth-VirtualBox:~$ free
              total        used        free      shared  buff/cache   available
Mem:      4430768       766228       3027772        7448       636768       3433216
Swap:      2097148           0       2097148
```

Carte mère et version du BIOS : avec la commande lshw pour avoir les informations sur la carte mère et la version du bios

```
srikanth@srikanth-VirtualBox:~$ sudo lshw
srikanth-virtualbox
description: Computer
product: VirtualBox
vendor: innotek GmbH
version: 1.2
serial: 0
width: 64 bits
capabilities: smbios-2.5 dmi-2.5 vsyscall32
configuration: family=Virtual Machine uuid=E6C27E0B-E74E-5B4C-89D9-DD45BB80
91B9
*-core
description: Motherboard
product: VirtualBox
vendor: Oracle Corporation
physical id: 0
version: 1.2
serial: 0
*-firmware
description: BIOS
vendor: innotek GmbH
physical id: 0
version: VirtualBox
date: 12/01/2006
size: 128KiB
capacity: 128KiB
capabilities: isa pci cdboot bootselect int9keyboard int10video acpi
```

Cartes additionnelles : pour les cartes additionnelle on utilise la commande suivante :lshw -class network

```
srikanth@srikanth-VirtualBox:~$ lshw -class network
WARNING: you should run this program as super-user.
*-network
description: Ethernet interface
product: 82540EM Gigabit Ethernet Controller
vendor: Intel Corporation
physical id: 3
bus info: pci@0000:00:03.0
logical name: enp0s3
version: 02
serial: 08:00:27:22:39:a2
size: 1Gbit/s
capacity: 1Gbit/s
width: 32 bits
clock: 66MHz
capabilities: bus_master cap_list ethernet physical tp 10bt 10bt-fd 100b
t 100bt-fd 1000bt-fd autonegotiation
configuration: autonegotiation=on broadcast=yes driver=e1000 driverversi
on=5.11.0-27-generic duplex=full ip=10.0.2.15 latency=64 link=yes mingnt=255 mu
lticast=yes port=twisted pair speed=1Gbit/s
resources: irq:19 memory:f0200000-f021ffff ioport:d020(size=8)
WARNING: output may be incomplete or inaccurate, you should run this program as
super-user.
```

Partitionnement du disque dur : pour déterminer les partitionnement de disque on utilise la commande suivante : sudo fdisk -l /dev/sda

```
srikanth@srikanth-VirtualBox:~$ sudo fdisk -l /dev/sda
Disk /dev/sda: 58,39 GiB, 62684954624 bytes, 122431552 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x080f141c

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sda1   *        2048     1050623     1048576    512M  b W95 FAT32
/dev/sda2             1052670  122429439  121376770    57,9G  5 Extended
/dev/sda5             1052672  122429439  121376768    57,9G  83 Linux
srikanth@srikanth-VirtualBox:~$
```

- Comment lire le journal de démarrage du système (boot) ?
- Comment lire de manière continue le journal d'événement ?

Pour le journal de demmarage du système

```
srikanth@srikanth-VirtualBox:~$ last reboot
reboot    system boot    5.11.0-27-generi Tue Aug 31 20:15    still running
reboot    system boot    5.11.0-27-generi Tue Aug 31 14:09    still running
reboot    system boot    5.11.0-27-generi Tue Aug 31 09:15    still running

wtmp begins Tue Aug 31 09:15:15 2021
srikanth@srikanth-VirtualBox:~$
```

Pour lire de manière continue le journal d'événement

```
srikanth@srikanth-VirtualBox:~$ tail -f /var/log/syslog
Aug 31 10:12:05 srikanth-VirtualBox dbus-daemon[522]: [system] Activating via s
ystemd: service name='org.freedesktop.nm_dispatcher' unit='dbus-org.freedesktop
.nm-dispatcher.service' requested by ':1.13' (uid=0 pid=523 comm="/usr/sbin/Net
workManager --no-daemon " label="unconfined")
Aug 31 10:12:05 srikanth-VirtualBox systemd[1]: Starting Network Manager Script
 Dispatcher Service...
Aug 31 10:12:05 srikanth-VirtualBox dbus-daemon[522]: [system] Successfully act
ivated service 'org.freedesktop.nm_dispatcher'
Aug 31 10:12:05 srikanth-VirtualBox systemd[1]: Started Network Manager Script
 Dispatcher Service.
Aug 31 10:12:06 srikanth-VirtualBox NetworkManager[523]: <info> [1630397526.37
31] manager: NetworkManager state is now CONNECTED_GLOBAL
Aug 31 10:12:06 srikanth-VirtualBox whoopsie[680]: [10:12:06] The default IPv4
route is: /org/freedesktop/NetworkManager/ActiveConnection/1
Aug 31 10:12:06 srikanth-VirtualBox whoopsie[680]: [10:12:06] Not a paid data p
lan: /org/freedesktop/NetworkManager/ActiveConnection/1
Aug 31 10:12:06 srikanth-VirtualBox whoopsie[680]: [10:12:06] Found usable conn
ection: /org/freedesktop/NetworkManager/ActiveConnection/1
Aug 31 10:12:06 srikanth-VirtualBox whoopsie[680]: [10:12:06] online
Aug 31 10:12:16 srikanth-VirtualBox systemd[1]: NetworkManager-dispatcher.servi
ce: Succeeded.
```

- Ouvrez un terminal.

Dans quel répertoire vous trouvez-vous ?

on se trouve dans le dossier « ~ », qui signifie qu'on est dans notre dossier personnel. Donc mon dossier personnel sous linux est /home/srikanth

```
srikanth@srikanth-VirtualBox:~$ pwd
/home/srikanth
```

- Dans votre répertoire de connexion, créez un répertoire tmp
- Positionnez les droits d'accès à rwx r-x --- pour tmp
- Copiez les fichiers passwd, group et hosts du répertoire /etc dans tmp
- Changez le nom de hosts en hotes.
- Positionnez les droits d'accès à rw- r--- --- pour le fichier hotes. Lisez le contenu de hotes.

Remarque : la lecture du fichier ~/tmp/hotes est permise. Le fichier peut néanmoins être vide.

- Retirez au propriétaire le droit en lecture sur le fichier hotes et essayez de le lire.

Quel est la signification du message d'erreur obtenu ?

Le message signifie que la permission nous a été refusée

```

srikanth@srikanth-VirtualBox:~/tmp$ ls -l
total 12
-rw-r--r-- 1 srikanth srikanth 1087 août 31 11:05 group
-rw-r----- 1 srikanth srikanth 234 août 31 11:07 hotes
-rw-r--r-- 1 srikanth srikanth 2813 août 31 11:01 passwd
srikanth@srikanth-VirtualBox:~/tmp$ chmod 240 hotes
srikanth@srikanth-VirtualBox:~/tmp$ ls -l
total 12
-rw-r--r-- 1 srikanth srikanth 1087 août 31 11:05 group
--w-r----- 1 srikanth srikanth 234 août 31 11:07 hotes
-rw-r--r-- 1 srikanth srikanth 2813 août 31 11:01 passwd
srikanth@srikanth-VirtualBox:~/tmp$ cat hotes
cat: hotes: Permission denied
srikanth@srikanth-VirtualBox:~/tmp$ █

```

- Remettez pour le propriétaire le droit en lecture sur le fichier hotes.
- Retirez pour le propriétaire le droit en écriture sur le répertoire tmp.
- Essayez de détruire hotes.

Quel est la signification du message d'erreur obtenu ?

Le message nous dit que c'est impossible de supprimer, permission refuser

```

srikanth@srikanth-VirtualBox:~$ ls -l
total 40
drwxr-xr-x 2 srikanth srikanth 4096 août 31 09:20 Desktop
drwxr-xr-x 2 srikanth srikanth 4096 août 31 09:20 Documents
drwxr-xr-x 2 srikanth srikanth 4096 août 31 09:20 Downloads
-rw-r--r-- 1 srikanth srikanth 1087 août 31 10:54 etc
drwxr-xr-x 2 srikanth srikanth 4096 août 31 09:20 Music
drwxr-xr-x 2 srikanth srikanth 4096 août 31 09:20 Pictures
drwxr-xr-x 2 srikanth srikanth 4096 août 31 09:20 Public
drwxr-xr-x 2 srikanth srikanth 4096 août 31 09:20 Templates
drwxr-x-- 2 srikanth srikanth 4096 août 31 11:10 tmp
drwxr-xr-x 2 srikanth srikanth 4096 août 31 09:20 Videos
srikanth@srikanth-VirtualBox:~$ chmod 550 tmp
srikanth@srikanth-VirtualBox:~$ cd /etc/
srikanth@srikanth-VirtualBox:/etc$ cd tmp
bash: cd: tmp: No such file or directory
srikanth@srikanth-VirtualBox:/etc$ cd
srikanth@srikanth-VirtualBox:~$ cd tmp
srikanth@srikanth-VirtualBox:~/tmp$ ls
group hotes passwd
srikanth@srikanth-VirtualBox:~/tmp$ rm hotes
rm: cannot remove 'hotes': Permission denied
srikanth@srikanth-VirtualBox:~/tmp$ █

```

- Retirez pour le propriétaire le droit en lecture sur le répertoire tmp.
- Essayez de lister le contenu de tmp.

Quel est la signification du message d'erreur obtenu ?

Le message nous , qu'on ne peut pas accéder au directory , permission refuser

```
srikanth@srikanth-VirtualBox:~/tmp$ cd
srikanth@srikanth-VirtualBox:~$ chmod 150 tmp
srikanth@srikanth-VirtualBox:~$ cd tmp
srikanth@srikanth-VirtualBox:~/tmp$ ls
ls: cannot open directory '.': Permission denied
srikanth@srikanth-VirtualBox:~/tmp$
```

- Lisez le contenu de hotes.

Pourquoi pouvez-vous le lire ?

Oui on peut lire le contenu, car le propriétaire possède les droit a la lecture

```
srikanth@srikanth-VirtualBox:~/tmp$ cat hotes
127.0.0.1      localhost
127.0.1.1      srikanth-VirtualBox

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
srikanth@srikanth-VirtualBox:~/tmp$
```

- Retirez pour le propriétaire le droit en exécution sur le répertoire tmp.
- Essayez de vous positionner sur ce répertoire.
- Quel est la signification du message d'erreur obtenu ?

Le message signifie qu'on essaye d'accéder a un directory dans laquelle on n'a pas la permission

```
srikanth@srikanth-VirtualBox:~$ chmod 050 tmp
srikanth@srikanth-VirtualBox:~$ cd tmp
bash: cd: tmp: Permission denied
srikanth@srikanth-VirtualBox:~$
```

- Essayez de lire le contenu de hotes.

Quel est la signification du message d'erreur obtenu ?

Le message nous dit permission refuser, pour lire le fichier hôtes

```
srikanth@srikanth-VirtualBox:~$ cat /home/srikanth/tmp/hotes
cat: /home/srikanth/tmp/hotes: Permission denied
srikanth@srikanth-VirtualBox:~$
```


6 Source, code assembleur, binaire et débogage.

- Lisez la fiche « rappel compilation » ;
- Réalisez le premier programme de la fiche welcome.c ;
- Lancez et analysez la compilation étape par étape comme décrite dans la fiche « rappel compilation ».

Copier ici, le **code assembleur** de ce programme.

Voici le code assembleur

```
srikanth@srikanth-VirtualBox:~/Documents$ gcc -E welcome.c > welcome.i
srikanth@srikanth-VirtualBox:~/Documents$ gcc -S welcome.i
srikanth@srikanth-VirtualBox:~/Documents$ gcc welcome.s
srikanth@srikanth-VirtualBox:~/Documents$ cat welcome.s
        .file     "welcome.c"
        .text
        .section   .rodata
.LC0:
        .string   "Rex is welcome you!"
        .text
        .globl    main
        .type     main, @function
main:
.LFB0:
        .cfi_startproc
        endbr64
        pushq     %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq      %rsp, %rbp
        .cfi_def_cfa_register 6
        leaq      .LC0(%rip), %rdi
        movl      $0, %eax
        call      printf@PLT
        movl      $0, %eax
        popq      %rbp
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
```

La suite du code :

```

.LFE0:
    .size    main, .-main
    .ident   "GCC: (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0"
    .section .note.GNU-stack,"",@progbits
    .section .note.gnu.property,"a"
    .align 8
    .long    1f - 0f
    .long    4f - 1f
    .long    5
0:
    .string  "GNU"
1:
    .align 8
    .long    0xc0000002
    .long    3f - 2f
2:
    .long    0x3
3:
    .align 8
4:
srikanth@srikanth-VirtualBox:~/Documents$

```

Pour connaître le type d'un fichier (binaire, son, image, etc.), on peut se fier à son extension (.mp3, .jpeg, etc.). Cela est limité surtout quand le nom du fichier ne possède pas d'extension. La commande `file` sous Linux permet de connaître le type d'un fichier en faisant abstraction de son nom.

Quel type de fichier votre compilateur a-t-il généré ? Récupérez un fichier binaire exécutable pour Windows (Notepad++ par exemple) et testez la commande `file` avec celui-ci ?

Type du fichier binaire Linux : le type d'un fichier binaire linux est un fichier de type assembleur généré par le compilateur.

```

srikanth@srikanth-VirtualBox:~/Documents$ file welcome.s
welcome.s: assembler source, ASCII text
srikanth@srikanth-VirtualBox:~/Documents$

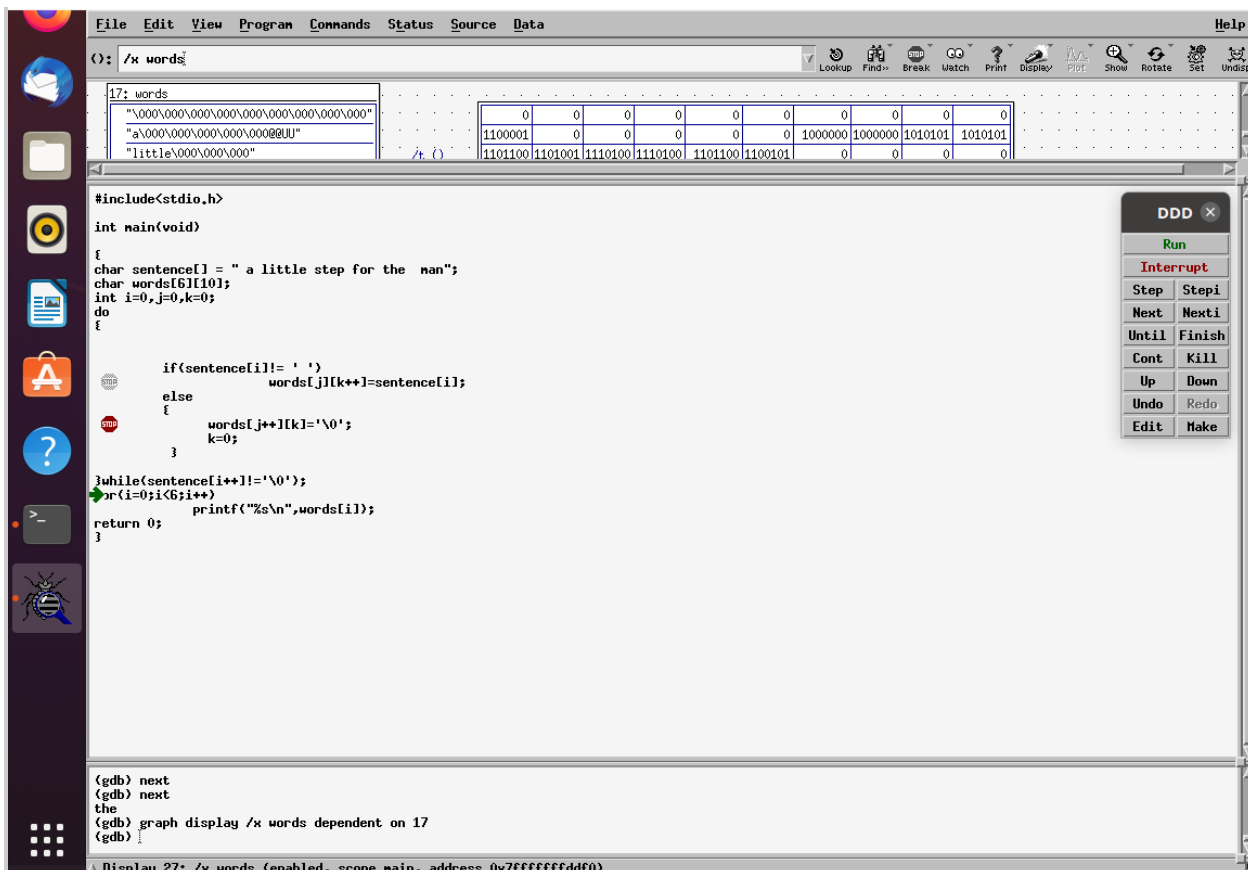
```

Type du fichier binaire Windows : le fichier binaire windows obtenue est un fichier de type ELF

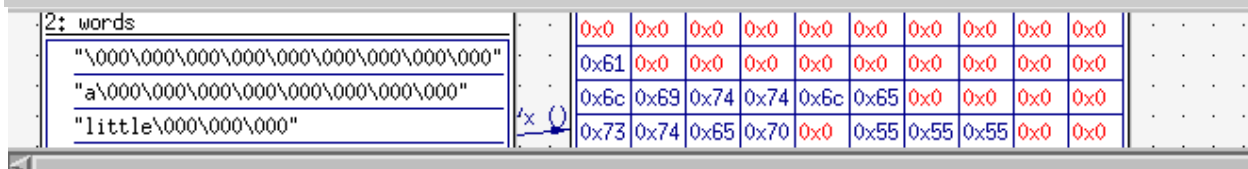

```
srikanth@srikanth-VirtualBox:~/Documents$ file welcome
welcome: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically li
nked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=7e54ec6262fa62ad8e
d34de44d2b68da74dc0652, for GNU/Linux 3.2.0, not stripped
srikanth@srikanth-VirtualBox:~/Documents$
```

- Analysez et écrivez le deuxième programme sentence2words.c ;
- Lancez-le pas à pas dans le débogueur ddd pour suivre l'évolution du chargement du tableau de mots.

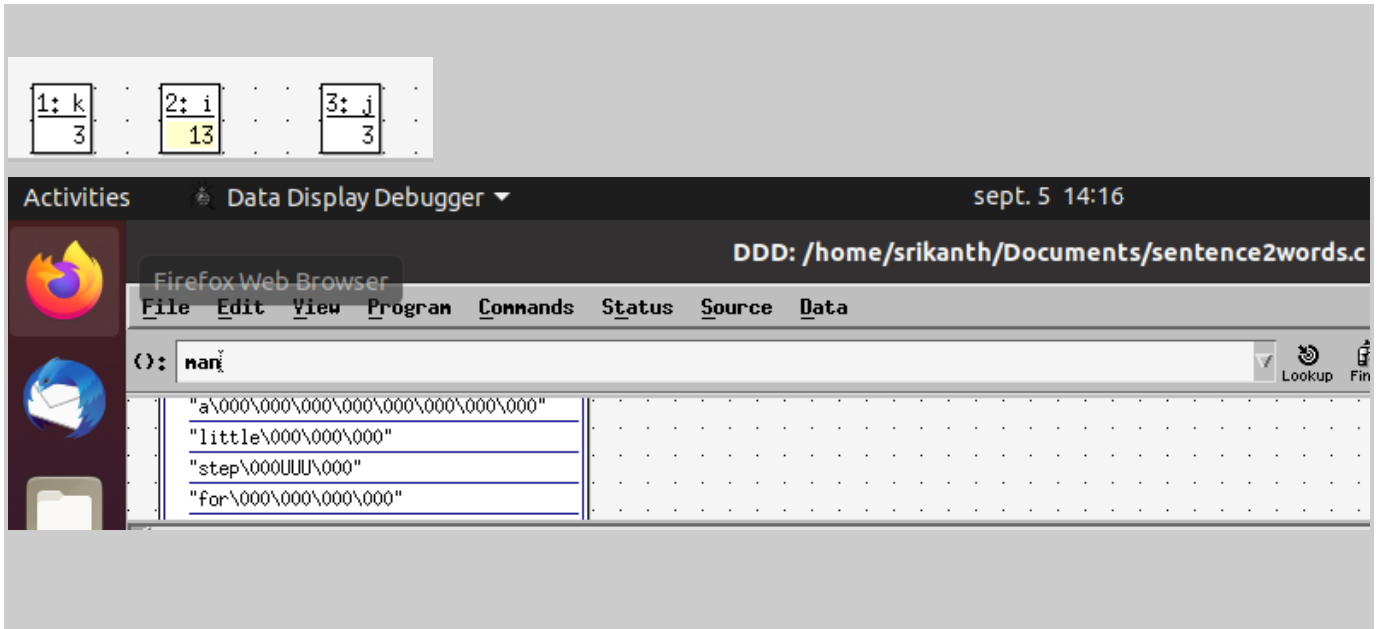
Réalisez une copie d'écran du débogueur affichant la zone mémoire du tableau rempli. Dans cette copie d'écran, on doit aussi voir la fenêtre de résultat du programme affichant le tableau de mots.



Affichage en hexa



```
sentence = " a little step for the man"
words    = [.,.,.]
i        = 13
j        = 3
k        = 3
```



7 Évolution de code

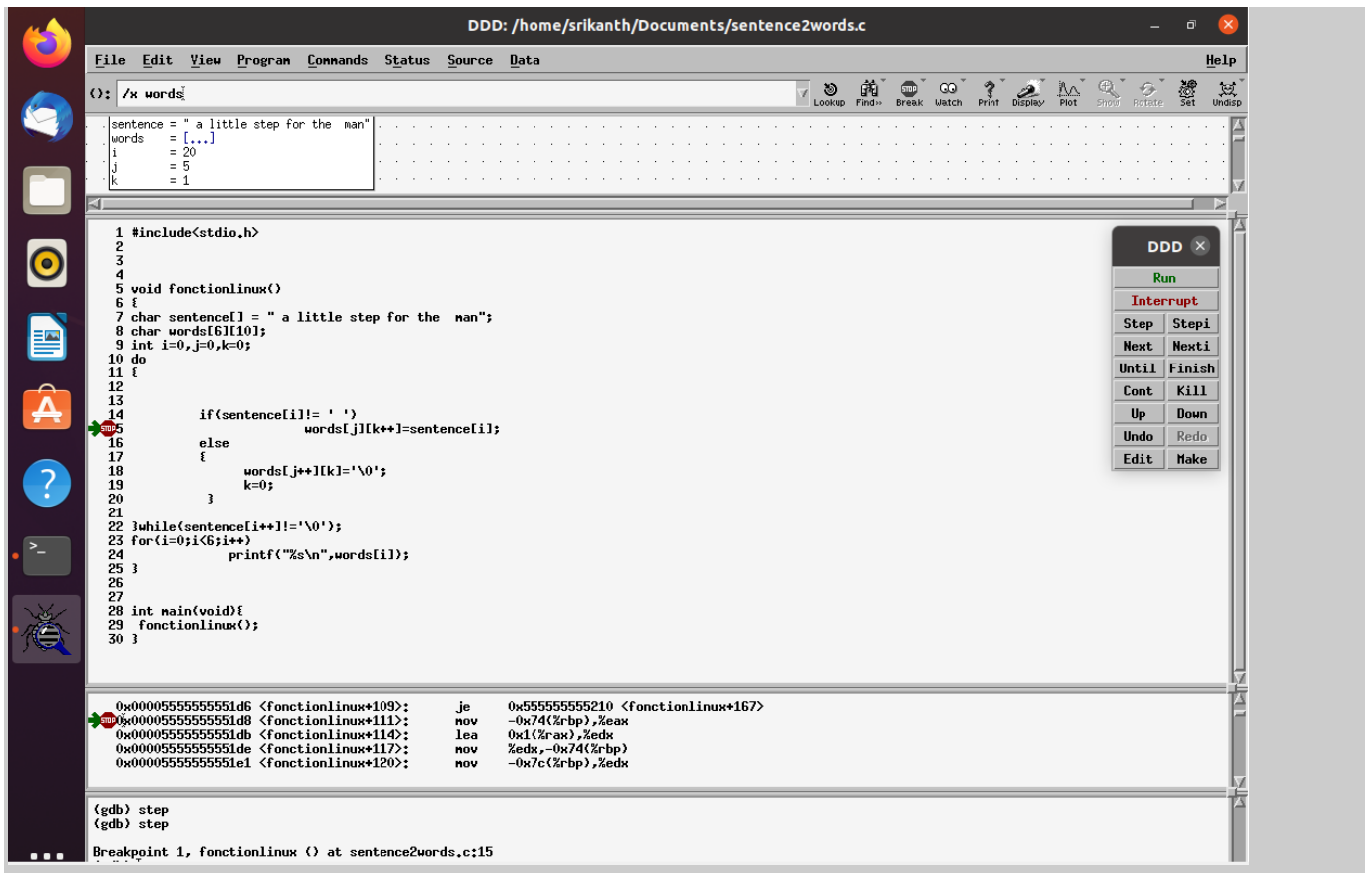
Proposez l'évolution suivante du code du deuxième programme :

- Au lancement du programme, la phrase initiale est demandée à l'utilisateur ;
- Le tableau est dimensionné dynamiquement en fonction de cette phrase ;
- L'affichage du tableau est réalisé par l'appel d'une fonction.

Copier votre code sur Moodle sous le nom de fichier « LAB0.c ».

Rappel : un fichier source qui ne compile pas n'est pas corrigé !

Copiez ci-dessous une copie d'écran du résultat de son exécution



a\000\000\000\000\000\000\000\000	0xb6	0xb9	0x74	0x74	0xb6	0xb5	0x0	0x0	0x0	0x0
"little\000\000\000"	0x73	0x74	0x65	0x70	0x0	0x55	0x55	0x55	0x0	0x0
"step\000UUU\000"	0x66	0x6f	0x72	0x0	0x0	0x0	0x0	0xc2	0x0	
"for\000\000\000\000"	0x74	0x0	0x0	0x0	0x0	0x0	0x47	0xde	0xff	0xff

3: i	4: j	5: k
20	5	1

```
6: sentence
" a little step for the man"
```

```
sentence = " a little step for the man" .
words    = [...]
i        = 20
j        = 5
k        = 2
```

Avec le débogueur on voit bien les variations de valeur quand on appuie sur step.

(le programme avec l'appel de fonction est à retrouver dans le fichier LAB0.C

8 Amélioration de la fiche « débogage »

L'objectif est d'améliorer cette fiche, exploitez ddd sur un programme utilisant des fonctions et expliquez le rôle des fichiers core.

Copiez votre fiche « débogage » sur Moodle en même temps que ce LAB

Le rôle des fichiers core : un fichier core est généré automatiquement si un programme ou un processus échoue.

Pour la fiche debogage, j'ai utiliser ce site : <https://stackoverflow.com/questions/5125896/how-to-disassemble-a-binary-executable-in-linux-to-get-the-assembly-code>

Vérifiez que vous avez bien copié 3 fichiers sur Moodle : ce LAB, le code de l'évolution du programme sentence2words.c et votre fiche débogage.