


---

# TP 2 VARI 1

Informations techniques PC Suse :

- (a) Pour démarrer une session : utilisateur **licencep** et mot de passe **7002n\***. Vous trouverez :
    - un gestionnaire de fichiers en haut à gauche placé dans le dossier personnel HOME
    - une icône lézard  en haut à droite pour accéder au menu.
  - (b) Pour démarrer *Processing* : clic sur l'icône lézard en haut à droit → Développement → Processing.
  - (c) Pour démarrer une *console* : clic sur l'icône lézard en haut à droit → Terminal → Konsole.
  - (d) Pour ouvrir un gestionnaire/navigateur de fichiers : clic sur l'icône lézard → Utilitaires → Dolphin.
  - (e) Pour lancer une commande : Alt + Space
  - (f) Pour modifier un fichier, clic droit sur le fichier → Ouvrir avec KWrite (ou autre éditeur de votre choix).
- 

## 1 Le Shell du système Linux : vitesse et mémoire

Démarrer une console/terminal en suivant les instructions au point (c) ci-dessus.

**Exercice 1** Taper la commande suivante dans le terminal *Shell*, en faisant attention aux espaces.

```
time wget cedric.cnam.fr/~porumbed/1giga
```

Elle télécharge un fichier d'1 GB $\approx$ 1000MB (commande `wget`) et affiche le temps utilisé pour cela, voir rubrique `real` (affiché par la commande `time` devant). **Calculer la vitesse (débit) du réseau**  $\frac{1000\text{MB}}{\text{temps (sec)}}$ , ex., si cela prend 9 secondes on a  $\frac{1000}{9}=111\text{MBs}$ . Vous pouvez faire le calcul avec une calculette, un programme `processing` ou taper dans le terminal une commande comme : `irb<<<"1000/9"`. Attention : le débit dépend de la charge du réseau (qui peut augmenter si vous êtes plusieurs à lancer cette commande en même temps).

**Exercice 2** Exécuter la commande suivante pour faire une copie du fichier. Déterminer la vitesse du disque dur.

```
time cp 1giga 1gigacopie
```

**Exercice 3** Taper la première commande ci-dessous. Noter la taille totale de la RAM. Remarquer la taille du SWAP. La deuxième commande présente les mêmes informations d'une manière encore plus détaillée.

```
free -h  
cat /proc/meminfo
```

**Exercice 4** Taper la commande `top`. Remarquer les processus qui consomment le plus de CPU. Taper `M` (attention : majuscule obligatoire) pour afficher les processus qui consomment le plus de mémoire RAM. Pour quitter, appuyer sur `q` ou `CTRL-C`.

**Exercice 5** Taper la commande suivante pour changer le gestionnaire de fenêtres et utiliser `icewm`.

```
icewm --replace
```

Appuyer `ALT-TAB` plusieurs fois pour observer les différentes fenêtres. Taper `CTRL-C` pour arrêter. Lancer la commande suivante pour revenir au gestionnaire de fenêtres original.

```
kwin_x11 --replace &
```

**Exercice 6** Télécharger une ou plusieurs machines virtuelles disponibles sur le site web :

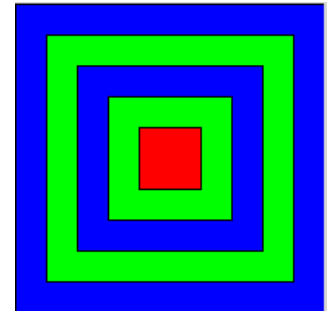
```
http://cedric.cnam.fr/~porumbed/vari1/vbox/
```

Télécharger des machines virtuelles (fichiers `vdi`) dans le dossier personnel. Faire tourner les différents systèmes d'exploitation. Vous pouvez utiliser `AntixLinux avec Processing` pour faire les prochains exercices `Processing` !

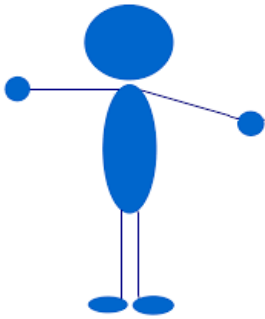
## 2 Processing

**Exercice 1** Soit le code ci-dessous. Ajouter deux lignes pour dessiner un nouveau carré rouge à l'intérieur du plus petit carré. L'objectif est d'obtenir un dessin similaire à celui en bas à droite.

```
1 size(200,200);
2 fill(0,0,255); //0 rouge, 0 vert, 255 bleu
3 rect(0,0,200,200); //taille rectangle 200X200
4 fill(0,255,0);
5 rect(20,20,160,160);
6 fill(0,0,255);
7 rect(40,40,120,120);
8 fill(0,255,0);
9 rect(60,60,80,80);
```



**Exercice 2** Utiliser quatre appels `line(...)` et cinq appels `ellipse(...)` pour réaliser le dessin ci-après.



Indication :

- utiliser `fill(r,g,b)` pour choisir la couleur de l'intérieur des ellipses
- utiliser `stroke(r,g,b)` pour choisir les couleurs des lignes et des contours des ellipses

**Exercice 3** Écrire un programme qui affiche “carré” si  $x = y^2$  ou  $y = x^2$ , où  $x$  et  $y$  sont deux variables saisies en dur au début du programme.

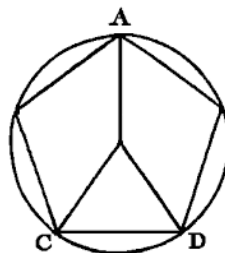
**Indication :** vous pouvez utiliser deux `if` séparés. Le code en bas à gauche affiche « salut » si  $x = 8$  ou  $x = 9$ . Sinon, il affiche rien. Le code en bas à droite mène au même résultat que celui de gauche.

```
if(x==8){
    println("salut");
}
if(x==9){
    println("salut");
}
```

```
if((x==8)|| (x==9)){
    println("salut");
}
```

**Exercice 4** Utiliser 5 appels `line(x1,y1,x2,y2)` pour dessiner un pentagone. Écrire “ABCD” à l'intérieur du pentagone avec la fonction `text(chaineDeCaractères,posX,posY)`.

**Exercice 5** Utiliser plusieurs appels `line(...)` ou `ellipse(...)` pour réaliser le dessin à droite.



**Exercice 6** BONUS Soit une variable *an* initialisée au début du programme. Afficher le nombre de jours du mois de février de cet an. On rappelle qu'une année est bissextile (29 jours en février) si

- l'année est divisible par 4 et non divisible par 100, ou
- l'année est divisible par 400.

Indication 1 : utiliser `if(a%4==0)...` pour vérifier si la variable *a* est divisible par 4.

Indication 2 : utiliser `if((Cond1)|| (Cond2))` pour vérifier si “(Cond1 ou Cond2)” est vraie.

Indication 3 : utiliser `if(((Cond1)&&(Cond2))|| (Cond3))` pour tester si “((Cond1 et Cond2) ou Cond3)” est vraie.

Indication 4 : utiliser `if(a!=0) ..` pour une conditionnelle  $a \neq 0$ .