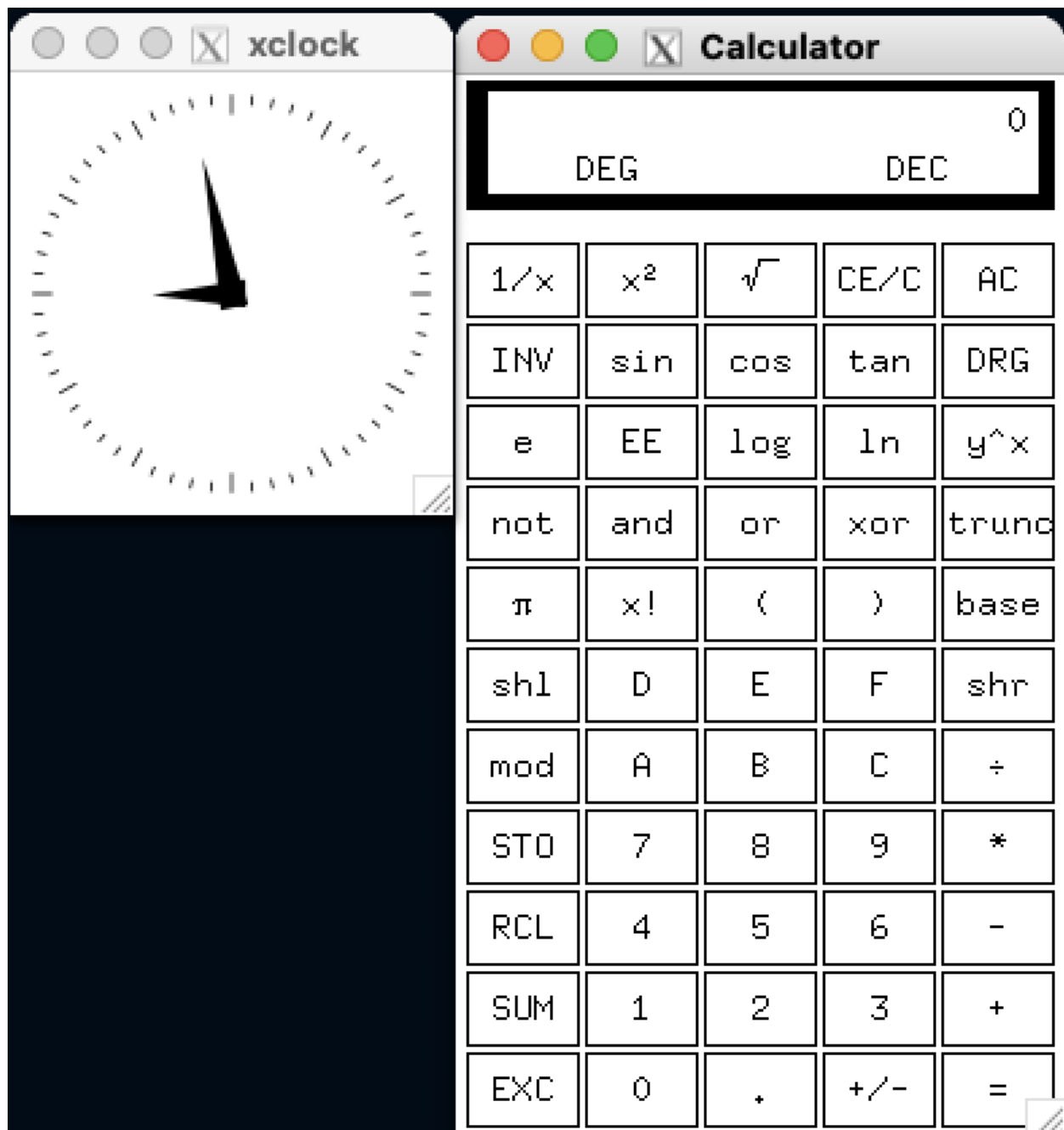


TP3 KAUFFMANN // BRIENT

Exercise 1.

A.



```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os master ± xcalc &; xclock &
[1] 15832
[2] 15833
```

B. On peut voir que un zsh en plus est apparu dans le retour de la commande ps, cela est arrivélors du lancement du second terminal

```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os master ± ps
  PID TTY          TIME CMD
  7783 ttys000      0:00.30 -zsh
 15832 ttys000      0:00.05 xcalc
 15833 ttys000      0:00.05 xclock
  6847 ttys001      0:00.12 -zsh
  7431 ttys001      0:00.01 tmux
  7997 ttys002      0:00.19 -zsh
 16450 ttys002      0:00.32 vim Rendu
 10544 ttys003      0:00.16 -zsh
```

C. uid: ID de l'utilisateur ayant lancé le processus

pri: Organisation de la priorité

vsz=SZ: Taille virtuel en Kbytes

tty: Nom complet du terminal ayant lancé le processus

time: Temps CPU accumulé : Utilisateur + Système

```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os master ± ps -l
  UID  PID  PPID      F CPU PRI NI       SZ     RSS WCHAN    S        ADDR  TTY          TIME CM
D
  602  7783  7433    4006   0  31   0 34189696  3204 -      Ss        0  ttys000    0:00.30 -
  602 15832  7783    4046   0  31   5 34153148  1844 -      SN        0  ttys000    0:00.06 x
  602 15833  7783    4046   0  31   5 34195956  7376 -      SN        0  ttys000    0:00.06 x
  602  6847  6846    4006   0  31   0 34189584  2004 -      S         0  ttys001    0:00.12 -
  602  7431  6847    4006   0  31   0 34151228   976 -      S+        0  ttys001    0:00.01 t
  602  7997  7433    4006   0  31   0 34189696  2344 -      Ss        0  ttys002    0:00.19 -
  602 16450  7997    4006   0  31   0 34211968 15876 -      S+        0  ttys002    0:01.55 v
  602 10544  7433    4006   0  31   0 34189696  2532 -      Ss+       0  ttys003    0:00.16 -
```

D. "ps -e" permet de voir tout les processus y compris ceux qui ne sont pas lancée par l'utilisateur courant et aussi les processus qui n'ont pas été lancée avec un terminal.

```

user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP03 master ± ps -e
PID TTY TIME CMD
1 ?? 2:40.07 /sbin/launchd
95 ?? 1:47.82 /usr/libexec/logd
96 ?? 0:38.71 /usr/libexec/UserEventAgent (System)
98 ?? 0:00.86 /Applications/ESET Endpoint Security.app/Contents/MacOS/esets_ctl
100 ?? 0:01.69 /System/Library/PrivateFrameworks/Uninstall.framework/Resources/uninstalld
101 ?? 1:56.28 /System/Library/PrivateFrameworks/CoreServices.framework/Versions/A/Frameworks/FSEvents.framework/Versions/A/Support/fsevents
102 ?? 0:13.74 /System/Library/PrivateFrameworks/MediaRemote.framework/Support/mediaremoted
104 ?? 1:02.41 /usr/sbin/systemstats --daemon
106 ?? 1:40.74 /usr/libexec/configd
107 ?? 0:00.20 endpointsecurityd
108 ?? 1:01.25 /System/Library/CoreServices/powerd.bundle/powerd
113 ?? 0:10.73 /usr/libexec/remoted
115 ?? 0:03.16 /usr/libexec/keybagd -t 15
116 ?? 0:00.26 /System/Library/PrivateFrameworks/MobileSoftwareUpdate.framework/Support/softwareupdated
118 ?? 0:01.88 /usr/libexec/watchdogd
122 ?? 2:07.74 /System/Library/Frameworks/CoreServices.framework/Frameworks/Metadata.framework/Support/nds
123 ?? 0:41.88 /Library/Application Support/Fortinet/Forticlient/bin/ztnafw
124 ?? 0:00.12 /System/Library/CoreServices/iconservicesd
125 ?? 0:04.27 /usr/libexec/kernelmanagerd
126 ?? 0:04.22 /usr/libexec/diskarbitrationd
129 ?? 1:05.68 /usr/libexec/coreduetd
130 ?? 0:10.90 /usr/sbin/syslogd
133 ?? 0:00.03 /usr/libexec/thermalmonitord

```

E. Pour obtenir plus d'informations, il suffit d'utiliser l'option "-v" avec ps.

```

user@AB-MBP-008 ~/Documents/EfreiL3/unix-os master ± ps -vl
PID STAT TIME SL RE PAGEIN VSZ RSS LIM TSIZ %CPU %MEM COMMAND UID PPID F CPU PRI NI WCHAN ADDR TTY
37867 S+ 0:05.43 0 0 0 34213020 23448 - 0 0,0 0,3 vim Rendu 602 7997 4006 0 31 0 - 0 ttys002
15833 SN 0:00.07 0 0 0 34195956 7404 - 0 0,0 0,1 xclock 602 7783 4046 0 31 5 - 0 ttys000
7783 Ss 0:00.34 0 0 0 34189696 3356 - 0 0,3 0,0 -zsh 602 7433 4006 0 31 0 - 0 ttys000
7997 Ss 0:00.23 0 0 0 34189696 3160 - 0 0,0 0,0 -zsh 602 7433 4006 0 31 0 - 0 ttys002
10544 Ss+ 0:00.22 0 0 0 34189696 3068 - 0 0,0 0,0 -zsh 602 7433 4006 0 31 0 - 0 ttys003
6847 S 0:00.12 0 0 0 34189584 2004 - 0 0,0 0,0 -zsh 602 6846 4006 0 31 0 - 0 ttys001
15832 SN 0:00.06 0 0 0 34153148 1844 - 0 0,0 0,0 xcalc 602 7783 4046 0 31 5 - 0 ttys000
7431 S+ 0:00.01 0 0 0 34151228 976 - 0 0,0 0,0 tmux 602 6847 4006 0 31 0 - 0 ttys001

```

F. La commande permettant de savoir les processus lancé par root est "ps -u 0"

Exercice 2.

A. On peut remarquer que le PID change car le programme est relancé à chaque fois.

```

user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP03 master ± ./get_pid.out
Le pid est : 62217%
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP03 master ±

```

B. On peut remarquer que le PPID ne change pas car c'est le PID du processus parent donc notre terminal.

```

user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP03 master ± ./get_pid.out
Le pid est : 65242
Le ppid est : 10544%
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP03 master ± ./get_pid.out
Le pid est : 65434
Le ppid est : 10544%
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP03 master ± ./get_pid.out
Le pid est : 65460
Le ppid est : 10544%
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP03 master ±

```

Exercice 3.

A.

```
> ./exo3.out
Je suis le père numéro 8670
Je suis le fils numéro 8671 du père 8670
Je suis le fils numéro 8672 du père 8670
Je suis le fils numéro 8673 du père 8670
Je suis le petit fils numéro 8674 du père 8671
Je suis le peit fils numéro 8675 du père 8671
> ./exo3.out
Je suis le père numéro 8690
Je suis le fils numéro 8691 du père 8690
Je suis le fils numéro 8692 du père 8690
Je suis le fils numéro 8693 du père 8690
Je suis le petit fils numéro 8694 du père 8691
Je suis le peit fils numéro 8695 du père 8691
> vim rendu
~/EfreiL3/u/TP03 | master ?2 | ok | 45s | 14:46:27
```

B. On peut observer quand on lance plusieurs fois le programme est que les pid changent car de nouveaux processus sont créés à chaque lancement du programme.

Exercice 4.

```

exo3.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/types.h>
4  #include <unistd.h>
5  #include <sys/wait.h>
6
7  int main(void){
8      printf("Je suis le père numéro %d\n", getpid());
9      if(fork() == 0){
10
11
12          if(fork() == 0){
13              printf("Je suis le petit fils numéro %d du père %d\n", getpid(), getppid());
14          }
15          else{
16              if(fork() == 0){
17                  printf("Je suis le petit fils numéro %d du père %d\n", getpid(), getppid());
18              }
19              else{
20                  sleep(1); //A
21                  printf("Je suis le fils 1 numéro %d du père %d\n", getpid(), getppid());
22              }
23          }
24      }
25      else{
26          if(fork() == 0){
27              sleep(1); //B
28              printf("Je suis le fils 2 numéro %d du père %d\n", getpid(), getppid());
29          }
30          else{
31              if(fork() == 0){
32                  sleep(1); //A
33                  printf("Je suis le fils 3 numéro %d du père %d\n", getpid(), getppid());
34              }
35          }
36      }
37  }
38  while(wait(NULL) > 0);
39  }
40

```

Snipped

```

> ./exo3.out
Je suis le père numéro 19632
Je suis le petit fils numéro 19636 du père 19633
Je suis le petit fils numéro 19637 du père 19633
Je suis le fils 2 numéro 19634 du père 19632
Je suis le fils 3 numéro 19635 du père 19632
Je suis le fils 1 numéro 19633 du père 19632

```

~/EfreiL3/u/TP03 | master ?6

ok | 19:21:41

Exercice 5.

```
Essayez 'ps --aide <simple|liste|sortie|threads|diver
s|tous>'
ou 'ps --aide <s|l|o|h|d|t>'
pour plus d'aide.

Pour plus de détails, consultez ps(1).
Child number 17831 exited
Child number 17830 exited first%
~/EfreiL3/u/TP03 | master ?2 | ok | 15:55:59
```

ex05.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sys/types.h>
4  #include <unistd.h>
5  #include <sys/wait.h>
6
7  int main(void){
8      pid_t child_state;
9      pid_t child_state2;
10     pid_t rc_pid;
11     pid_t rc_pid2;
12     int status;
13     if(fork() == 0){
14         execlp("ls","ls","-l","/",NULL);
15     }
16     else{
17         if(fork() == 0){
18             execlp("ps","ps","-l","/",NULL);
19         }
20         else{
21             rc_pid = waitpid(getpid()+1, &child_state, 0);
22             if(rc_pid > 0){
23                 if(WIFEXITED(child_state)){
24                     printf("Child number %d exited\n", getpid()+1);
25                     status=1;
26                 }
27             }
28             rc_pid2 = waitpid(getpid()+2, &child_state2, 0);
29             if(rc_pid2 > 0){
30                 if(WIFEXITED(child_state2)){
31                     printf("Child number %d exited\n", getpid()+2);
32                     status=0;
33                 }
34             }
35             if(!status)
36                 printf("Child number %d exited first", getpid()+1);
37             else
38                 printf("Child number %d exited first", getpid()+2);
39         }
40         while(wait(NULL) > 0);
41     }
42 }
```

snipped

