



# QCM – Signaux et Tubes

IF 110 – Introduction aux systèmes d'exploitations

# Les signaux sont une méthode de communications

- A. Par envoi de message
- B. Par partage de mémoire
- C. Ça dépend

#QDLE#Q#A\*BC#20#



# Les messages envoyés via des signaux sont

- A. Des données quelconques
- B. Encadrés par un intervalle de valeurs

#QDLE#Q#AB\*#15#

# Les signaux envoyés peuvent-être perdus ?

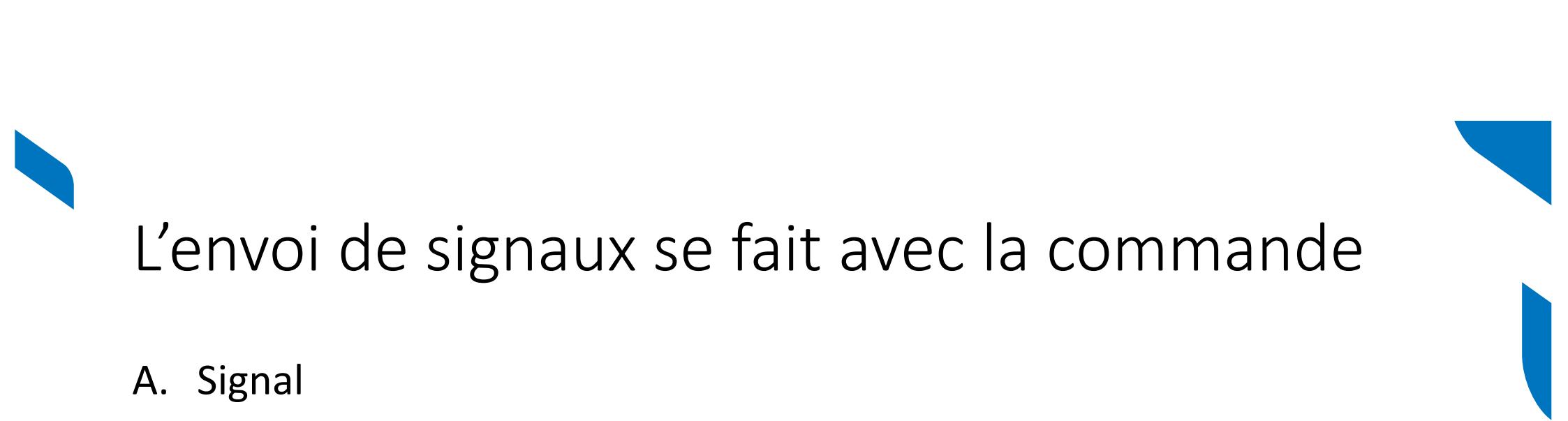
- A. Vrai
- B. Faux

#QDLE#Q#A\*B#10#

# L'ordre de reception des signaux est

- A. Le même que l'ordre d'émission
- B. Pas nécessairement le même que celui d'émission

#QDLE#Q#AB\*#15#



# L'envoi de signaux se fait avec la commande

- A. Signal
- B. Send
- C. &
- D. Ctrl+z
- E. Ctrl+c
- F. Kill

```
#QDLE#Q#ABCDEF*#20#
```

A la réception d'un signal, un processus va

- A. Executer automatiquement une routine/fonction définie par le développeur du programme
- B. Executer une routine/fonction par défaut
- C. Tuer le processus receiteur par défaut
- D. Ignorer le signal reçu par défaut
- E. Aucune de ces réponses

#QDLE#Q#AB\*C#25#

# Comment envoyer le signal SIGKILL ?

- A. Ctrl+z
- B. Ctr+c
- C. Kill -9 PID
- D. Kill –SIGSTOP PID
- E. Bg
- F. Aucune bonne réponse

#QDLE#Q#ABC\*DEF#20#

# Comment envoyer le signal SIGINT ?

- A. Ctrl+z
- B. Ctr+c
- C. Kill -9 PID
- D. Kill –SIGSTOP PID
- E. Bg
- F. fg

#QDLE#Q#AB\*CDEF#20#

# Comment redémarrer un processus ?

- A. Ctrl+z
- B. Ctr+c
- C. Kill -9 PID
- D. Kill –SIGSTOP PID
- E. Kill –SIGKILL PID
- F. Kill –SIGCONT PID
- G. Kill –USR1 PID

#QDLE#Q#ABCDEF\*G#20#

# Comment mettre en place un traitement/gestionnaire de signal

- A. Avec la commande trap à la fin du programme
- B. En recevant un signal approprié
- C. Avec la commande trap au début du programme
- D. Avec la commande kill au début du programme
- E. Aucune de ces réponses

#QDLE#Q#ABC\*DE#35#



Quel signal est envoyé par la commande  
« kill -9 »?

- A. SIGTERM
- B. SIGSTOP
- C. SIGKILL
- D. SIGINT

#QDLE#Q#ABC\*D#8#

# Que se passe-t-il lorsqu'un processus enfant se termine ?

- A. Il envoie un signal SIGCHLD à son parent
- B. Il attend que le système d'exploitation lui rende la main
- C. Le processus parent peut continuer son execution
- D. Il envoie SIGCHLD à son parent et attend que le système d'exploitation lui rende la main

#QDLE#Q#A\*BCD#30#



# Le signal envoyé à un processus via « Ctrl-Z » est

- A. SIGKILL
- B. SIGINT
- C. SIGTERM
- D. SIGCONT
- E. SIGSTOP

#QDLE#Q#A\*BCDE#15#

Si un utilisateur lance un processus via un terminal bash et tue ensuite ce terminal bash, que se passe-t-il pour ce processus ?

- A. Il devient orphelin mais et ensuite adopté par le processus 1
- B. Il devient orphelin et meurt
- C. Il meurt et devient un zombie
- D. Ces réponses n'ont aucun sens

#QDLE#Q#A\*BCD#40#

Un processus crée un processus enfant qui se termine. Le processus parent ne sait pas encore que son enfant a terminé. Ce processus enfant:

- A. Devient orphelin
- B. Est détruit
- C. Devient un zombie
- D. Ces réponses n'ont toujours aucun sens

#QDLE#Q#ABC\*D#40#

```
#!/bin/bash
```

p1.sh

```
Trap « echo Nope » SIGINT
echo pid: $$ 
while true; do
    sleep 1
done
```

```
$ ./p1.sh
pid: 12345
^Ctrl+c
```

## Que se passe-t-il ?

- A. Affichage de « nope »
- B. P1 se termine
- C. Aucune bonne réponse

#QDLE#Q#A\*BC#45#

23/10/2023

17

```

#!/bin/bash
echo pid: $$

while true; do
    sleep 1
done

trap << kill -SIGUSR1 $1 >> SIGUSR1
trap << kill -SIGUSR2 $1; exit >>SIGUSR2

```

p1.sh

```

#!/bin/bash
echo pid: $$

while true; do
    sleep 1
done

trap << ./p3.sh >> SIGUSR1
trap << exit >> SIGUSR2

```

p2.sh

```

#!/bin/bash
Echo «I've been summoned»

```

p3.sh

```

$ ./p2.sh &
pid: 12345
$ ./p1.sh 12345 &
pid: 12346
$ kill -SIGUSR1 12346

```

## Que se passe-t-il ?

- A. P3 a été lancé
- B. P1 et p2 continue de s'exécuter
- C. P1 se termine
- D. P1 et p2 deviennent des zombies
- E. Aucune bonne réponse

#QDLE#Q#AB\*CDE#90#

23/10/2023

18

```

#!/bin/bash
trap « kill -SIGUSR2 $1 » SIGUSR1 p1.sh
trap « kill -SIGUSR2 $1; exit » SIGUSR2
echo pid: $$ 
while true; do
    sleep 1
Done

```

```

#!/bin/bash
trap « echo hello » SIGUSR1 p2.sh
trap « ./p3.sh » SIGUSR2
echo pid: $$ 
while true; do
    sleep 1
done

```

```

#!/bin/bash
Echo «I've been summoned»

```

p3.sh

```

$ ./p2.sh &
pid: 12345
$ ./p1.sh 12345 &
pid: 12346
$ kill -SIGUSR2 12346

```

## Que se passe-t-il ?

- A. P1 terminé, p2 terminé, p3 n'a pas été lancé
- B. P1 continue, p2 continue, p3 n'a pas été lancé
- C. P1 terminé, p2 terminé, p3 a été lancé
- D. P1 continue, p2 continue, p3 a été lancé
- E. P1 terminé, p2 continue, p3 n'a pas été lancé
- F. P1 terminé, p2 continue, p3 a été lancé
- G. P1 continue, p2 terminé, p3 a été lancé

#QDLE#Q#ABCDEF\*G#120#