Université Bordeaux 1 — Licence Sciences & Technologie - Informatique
Utilisation d'Unix — 2009–2010

Test de connaissances n°1

Durée : 1h30
Tous documents autorisés. Les réponses doivent être écrites directement sur le sujet, dans les cadres prévus à cet effet. Votre voisin n'est pas un document. Vérifiez bien que votre sujet comporte 9 pages.

Nom : .....
Prénom : .....
Groupe : .....

Exercice 1 : Cochez les états possibles d'un processus Unix. (0.5 pts)

latent ..... ☐ — stoppé ..... ☐
en exécution ..... ☐ — en attente ..... ☐
prioritaire ..... ☐ — ininterruptible ..... ☐

Bonnes réponses : en exécution, stoppé, en attente.

Exercice 2 : Sous les systèmes du types Unix, quel est le login d'usage du super utilisateur ? (0.5 pts)

obiwan kenobi ..... ☐ — superuser ..... ☐
root ..... ☐ — admin ..... ☐

Bonne réponse : root

Exercice 3 : Que contient traditionnellement le répertoire `/usr/sbin` sous Unix ? (0.5 pts)

des bibliothèques de logiciels ..... ☐ — des programmes pour le super utilisateur .. ☐
des programmes pour les utilisateurs ..... ☐ — des fichiers de données ..... ☐

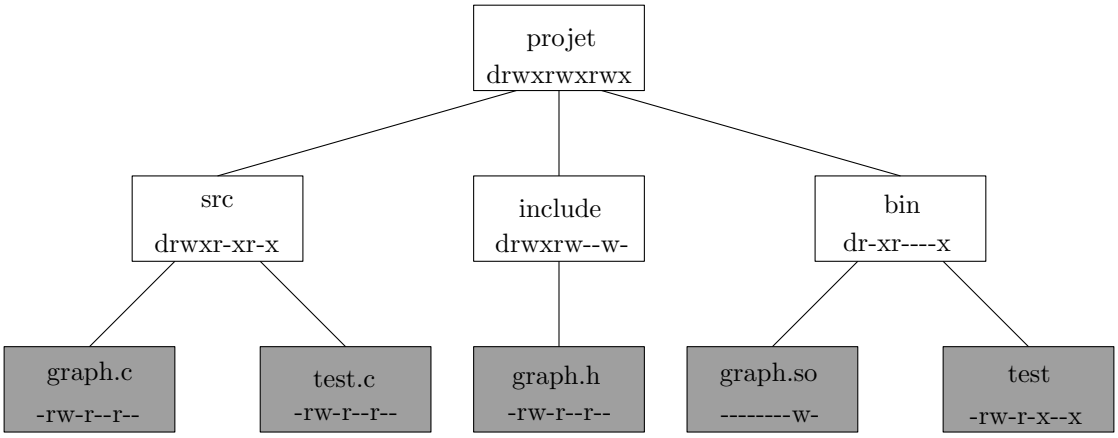
Bonne réponse : des programmes pour le super utilisateur

Exercice 4 : Tous les processus se voient attribué un PID unique par le noyau, il n'y a aucune exception : (0.5 pts)

Vrai ..... ☐ — Faux ..... ☐

Bonne réponse : Vrai

Exercice 5 : Arborescence (2 pts) On souhaite créer l'arborescence suivante. Les répertoires sont en blancs et les fichiers en gris. Les droits tels qu'affichés par la commandes **ls -l** sont indiqués en dessous du nom de chaque fichier et répertoire. Donner une suite de commandes créant cette arborescence avec les bons droits en supposant que l'on soit à l'intérieur d'un répertoire ayant les droits drwxr-xr-x .


projet
drwxrwxrwx
src
drwxr-xr-x
include
drwxrw--w-
bin
dr-xr----x
graph.c
-rw-r--r--
test.c
-rw-r--r--
graph.h
-rw-r--r--
graph.so
--------w-
test
-rw-r-x--x

```
mkdir projet
cd projet
mkdir src
cd src
touch graph.c
touch test.c
cd ..
mkdir include
cd include
touch graph.h
cd ..
chmod 762 include
mkdir bin
cd bin
touch graph.so
chmod 002 graph.so
touch test
chmod 651 test
cd ..
chmod 541 bin
chmod 777 .
```

Exercice 6 : Droits (2 pts) On considère l'arborescence de l'exercice précédent.

1) Cocher les opérations que l'utilisateur propriétaire de l'arborescence est autorisé à faire.
   - ☐ cat include/graph.h
   - ☐ ls include
   - ☐ touch include/vertex.h
   - ☐ ./bin/test
   - ☐ rm bin/graph.so
2) Cocher les opérations qu'un autre utilisateur dans le même groupe que l'utilisateur propriétaire de l'arborescence est autorisé à faire.
   - ☐ cat include/graph.h
   - ☐ ls include
   - ☐ touch include/vertex.h
   - ☐ ./bin/test
   - ☐ rm bin/graph.so
3) Cocher les opérations qu'un autre utilisateur dans un groupe différent que l'utilisateur propriétaire de l'arborescence est autorisé à faire.
   - ☐ cat include/graph.h
   - ☐ ls include
   - ☐ touch include/vertex.h
   - ☐ ./bin/test
   - ☐ rm bin/graph.so
4) Cocher les opérations qu'un super utilisateur est autorisé à faire.
   - ☐ cat include/graph.h
   - ☐ ls include
   - ☐ touch include/vertex.h

☐ ./bin/test

☐ rm bin/graph.so

1) 1,2,3 autorisées
2) 2 autorisée
3) 4 autorisées
4) 1,2,3,4,5 autorisées

Exercice 7 : Processus (3 pts) Voici le résultat de l'execution de la commande `ps` :

```
user100@trelawney :ps --User=user100 -F
UID        PID  PPID  C    SZ   RSS PSR STIME TTY          TIME CMD
user100  7859 20178  3  5352 14576   3 10:26 pts/22   00:00:04 emacs Server.c
user100 14421  7859  0  4374 14420   1 10:27 pts/6    00:00:00 /usr/X11R6/bin/gdb ...
user100 19529 20178  0   831   788   3 10:24 pts/22   00:00:00 man
user100 20121 20088  0  2958  2348   1 10:19 ?        00:00:01 sshd: user100@pts/22
user100 20178 20121  0  1489  2060   3 10:19 pts/22   00:00:00 bash
user100 29969 20178  0   944   904   1 10:28 pts/22   00:00:00 ps --User=user100 -F
(...)
```

Quel processus est lancé depuis Emacs ?

Le processus 14421. Ce processus correspond à la commande /usr/X11R6/bin/gdb –annotate=3

Quelle commande permet de tuer le processus 20178 ?

kill -9 20178

Qu'arrive-t-il au processus 7859 ?

Le processus 7859 étant un processus fils du processus 20178, ce processus est alors soit détruit, soit il reste en vie. Dans le deuxième cas, son processus parent devient alors le processus 1 (`init`).

Exercice 8 : Jobs (3 pts)

1) Rappeler les notions d'exécution en avant-plan et en arrière-plan. Comment faire passer un processus en avant-plan ? En arrière-plan ?
2) On effectue la série de commandes suivante :

```
(1) $ xeyes &
(2) $ emacs &
(3) $ firefox &
(4) $ jobs
(5) $ kill %3
(6) $ kill %1
(7) $ jobs
```

a) Qu'affiche le premier appel à la commande **jobs** (ligne 4) ?

```
[1]   Running                 xeyes &
[2]-  Running                 emacs &
[3]+  Running                 firefox &
```

b) Que font les commandes des lignes 5 et 6 ?

Elle tuent les processus xeyes et firefox.

c) Qu'affiche le deuxième appel à la commande **jobs** (ligne 7) ?

```
[2]+  Running                 emacs &
```

Exercice 9 : Alias (3 pts)

On effectue la série de commandes suivante :

```
 (1) $ cd /net/cremi/pbillon
 (2) $ alias rmdir='rm -rf'
 (3) $ alias uu='cd ~/tp/unix/'
 (4) $ alias ls='cal'
 (5) $ uu
 (6) $ mkdir 2009 2010
 (7) $ touch 2009/tp1 2009/tp2 2009/tp3
 (8) $ touch 2010/tp1 2010/tp2
 (9) $ ls 2009
(10) $ rmdir 2010
(11) $ ls 2010
(12) $ unalias ls
(13) $ ls 2010
```

1) Quel est le résultat de la commande de la ligne 9 ?

Affiche le calendrier de l'année 2009 car on a créé un alias ls sur la commande cal.

2) Donner le résultat de la commande de la ligne 11.

Affiche le calendrier de l'année 2010 car l'alias ls est toujours valide.

3) Qu'affiche l'exécution de la commande de la ligne 13 ?

Renvoit une erreur car on a supprimé l'alias ls à la ligne précédente. Ainsi la commande est censée afficher le contenu du répertoire 2010, or la commande de la ligne 10 a supprimé ce dossier et son contenu (l'alias rmdir sur la commande rm -rf a permis cette suppression.

Exercice 10 : Redirections (4 pts)

1) Rappeler la notion de « *pipe* » dans Unix. Quelle en est l'utilité ?

Le mécanisme appelé « *pipe* » est un opérateur permettant de connecter la sortie standard d'une premier programme à l'entrée standard d'un second sans devoir spécifier de fichier auxiliaire.

2) Comment obtenir le nombre de fichiers portant l'extension `*.txt` dans le répertoire `/opt/` (et dans tous ses sous-répertoires) ?

```
find /opt/ -name "*.txt" -print 2> /dev/null | wc -l
```

3) Que produit la commande `ps -U toto > proc ; wc -l proc ; rm proc` ?

Elle affiche le nombre total de processus exécuté par l'utilisateur toto sur la machine.

4) La commande précédente peut avoir un effet non désiré. Lequel ? Comment y remédier ?

Si le fichier proc existe dans le répertoire courant, il sera supprimé. Pour y remédier et éviter ainsi d'utiliser un fichier temporaire, le mécanisme de pipe semble être une bonne alternative. `ps -U toto | wc -l`

Exercice 11 : man (1 pts) Quels arguments doit-on passer à la commande `html2text` pour récupérer au format ascii dans le fichier `toto.txt` les pages d'accueils de l'université et de la fac.

```
html2text -o toto.txt -ascii http://www.u-bordeaux1.fr
ou
html2text -ascii http://www.u-bordeaux1.fr > toto.txt
```

```
NAME
       html2text - an advanced HTML-to-text converter

SYNOPSIS
       html2text -help
       html2text -version
       html2text  [ -unparse | -check ] [ -debug-scanner ] [ -debug-parser ] [
       -rcfile path ] [ -style ( compact | pretty ) ] [ -width width  ]  [  -o
       output-file ] [ -nobs ] [ -ascii ] [ input-url ...  ]

DESCRIPTION
       html2text  reads  HTML  documents  from the input-urls, formats each of
       them into a stream of plain text characters, and writes the  result  to
       standard  output (or into output-file, if the -o command line option is
       used).

(...)
       If  no  input-urls  are  specified on the command line, html2text reads
       from standard input. A dash as the input-url is  an  alternate  way  to
       specify standard input.

       html2text understands all HTML 3.2 constructs, but can render only part
       of them due to the limitations of the text output format. However,  the
       program attempts to provide good substitutes for the elements it cannot
       render.  html2text parses HTML 4 input, too, but not always as success-
       ful  as  other HTML processors. It also accepts syntactically incorrect
       input, and attempts to interpret it "reasonably".

 (...)
OPTIONS
       -ascii By default, html2text uses ISO 8859-1 for the output. Specifying
              this option, plain ASCII  is  used  instead.  To  find  out  how
              non-ASCII characters are rendered, refer to the file "ascii.sub-
              stitutes".

       -check This option is for diagnostic purposes:  The  HTML  document  is
              only  parsed and not processed otherwise. In this mode of opera-
              tion, html2text will report on parse  errors  and  scan  errors,
              which  it  does not in other modes of operation. Note that parse
              and scan errors are not fatal for html2text, but may cause  mis-
              interpretation  of the HTML code and/or portions of the document
              being swallowed.

       -debug-parser
              Let html2text report on the tokens being  shifted,  rules  being
              applied,  etc., while scanning the HTML document. This option is
              for diagnostic purposes.

       -debug-scanner
              Let html2text report on each lexical token scanned, while  scan-
              ning  the HTML document. This option is for diagnostic purposes.

       -help  Print command line summary and exit.

       -nobs  By default, html2text renders underlined letters with  sequences
              like  "underscore-backspace-character" and boldface letters like
              "character-backspace-character", which works fine when the  out-
```

```
              put is piped into more(1), less(1), or similar. For other appli-
              cations, or when redirecting the output into a file, it  may  be
              desirable not to render character attributes with such backspace
              sequences, which can be  accomplished  with  this  command  line
              option.

       -o output-file
              Write  the  output  to output-file instead of standard output. A
              dash as the output-file is an alternate way to specify the stan-
              dard output.

       -rcfile path
              Attempt to read the file specified in path as RC file.

       -style ( compact | pretty )
              Style  pretty  changes some of the default values of the format-
              ting parameters documented in html2textrc(5).  To find out which
              and how the formatting parameter defaults are changed, check the
              file "pretty.style". If this option is omitted, style compact is
              assumed as default.

       -unparse
              This  option  is  for diagnostic purposes: Instead of formatting
              the parsed document, generate HTML code, that is  guaranteed  to
              be  syntactically  correct.  If html2text has problems parsing a
              syntactically incorrect HTML document, this option may help  you
              to  understand what html2text thinks that the original HTML code
              means.

       -version
              Print program version and exit.

       -width width
              By default, html2text formats the HTML documents  for  a  screen
              width  of  79 characters. If redirecting the output into a file,
              or if your terminal has a width other than 80 characters, or  if
              you  just  want  to  get  an idea how html2text deals with large
              tables and different terminal widths, you may want to specify  a
              different width.
(...)

NAME
     cal, ncal - displays a calendar and the date of easter

SYNOPSIS
     cal [-3jmy] [[month] year]
     ncal [-jJpwy] [-s country_code] [[month] year]
     ncal [-Jeo] [year]

DESCRIPTION
     The cal utility displays a simple calendar in traditional format
     and ncal offers an alternative layout, more options and the date
     of easter.  The new format is a little cramped but it makes a
     year fit on a 25x80 terminal.  If arguments are not specified,
     the current month is displayed.

(...)

    A single parameter specifies the year (1 - 5875706) to be
    displayed; note the year must be fully specified: “cal 89” will
    not display a calendar for 1989.  Two parameters denote the month
    and year; the month is either a number between 1 and 12, or a full
```

```
    or abbreviated name as specified by the current locale.

     A year starts on Jan 1.

(...)

NAME
       ps - report a snapshot of the current processes.

SYNOPSIS
       ps [options]

DESCRIPTION
       ps displays information about a selection of the active
       processes. If you want a repetitive update of the selection and
       the displayed information, use top(1) instead.

(...)

PROCESS SELECTION BY LIST
       These options accept a single argument in the form of a
       blank-separated or comma-separated list. They can be used
       multiple times. For example: ps -p "1 2" -p 3,4

       -C cmdlist      Select by command name.
                       This selects the processes whose executable
                       name is given in cmdlist.


       -G grplist      Select by real group ID (RGID) or name.
                       This selects the processes whose real group
                       name or ID is in the grplist list. The real
                       group ID identifies the group of the user who
                       created the process, see getgid(2).

       U userlist      Select by effective user ID (EUID) or name.
                       This selects the processes whose effective user
                       name or ID is in userlist. The effective user
                       ID describes the user whose file access
                       permissions are used by the process (see
                       geteuid(2)). Identical to -u and --user.

       -U userlist     select by real user ID (RUID) or name.
                       It selects the processes whose real user name
                       or ID is in the userlist list. The real user ID
                       identifies the user who created the process,
                       see getuid(2).

       -g grplist      Select by session OR by effective group name.
                       Selection by session is specified by many
                       standards, but selection by effective group is
                       the logical behavior that several other
                       operating systems use. This ps will select by
                       session when the list is completely numeric (as
                       sessions are). Group ID numbers will work only
                       when some group names are also specified.
                       See the -s and --group options.

       p pidlist       Select by process ID. Identical to -p and --pid.

(...)
```