Makefile

Table des matières


```
#!/bin/bash

END="\033[0;39m"
b=0
while [ $b -le 1 ]
do
  d=0
  while [ $d -le 9 ]
    do
      u=0
      while [ $u -le 9 ]
      do
        nn=$d$u
        BEGIN='\033['${b}';'${nn}'m'
        u=$[u+1]

        echo -ne "${BEGIN}${nn}${END} "
      done
    d=$[d+1]
    echo -e "\n"
    echo -en "\\033[$[${d}*3]G"
    done
    echo -e "\n"
    b=$[b+1]
done

# echo -ne "\033[1;91m $(cowsay yé)\033[0;39m"
```



```
#!/bin/bash
# rq: le 'B' majuscule ne passe pas avec telnet :/

{
  echo "FAST_";
  echo "DEbUG EXIT";
  echo "ENDMSG";
  sleep 3;
} | telnet 192.168.1.167 1808 >/dev/null 2>&1
```



```
#!/bin/bash
#set -x

# pour le fun
```

```
poloAreponduOui="/tmp/poloAreponduOui.txt"

# %1: "texte a decouper"
# %2: 1,2 ou 3 pour début, milieu et fin
# %3: resultat
decoupage()
{
    tmp=$(echo "$1" | sed -e "s/\(.*\) - \(.*\)\(\.[[:alpha:]]*\)/\\$2/")

    if [ "$tmp" == "$1" ]
    then
echo "nom pas adapté a la substitution: $1"
exit 1
    fi

    eval $3=\'$tmp\'
}

if [ $# -lt 1 ]
then
    echo -ne "\nusage:\n"
    echo -ne "\t$ rm $poloAreponduOui\n"
    echo -ne "\t$ find . -name '?x*.avi' -exec ./rename.sh {} \;\n\n"
    exit 1
fi

# découpage (xxx - yyy.zzz)
decoupage "$1" 1 debut
decoupage "$1" 2 milieu1
decoupage "$1" 3 fin

# remplacer les espaces par des underscores
milieu2=$(echo $milieu1 | sed -e 's/ /_/g')
nouveauNom="${debut}-${milieu2}${fin}"

# a present c'est plus que du blabla...
echo -ne "mv\t'$1'\t'$nouveauNom'"

if [ -f $poloAreponduOui ]
then
    echo -ne "\n"
else
    read -p "\t? (y/n/!) : [n] " response
    case $response in
y|Y|yes|oui|svp|vasymongars)
    ;;
!|y!|yes!|toujours|alway|forever)
    touch $poloAreponduOui
    ;;
*)
    exit 0
    ;;
    esac
fi
```

```
# tout ça pour ça
mv "$1" "$nouveauNom"
```

- `$ man toto`

```
# vi toto.txt
# nroff toto.txt > /usr/share/man/man1/toto.1
# man toto
```

- menu.sh
- lpbook2

```
#!/bin/bash

rm -fr tmppdf
mkdir -p tmppdf
cp book.pdf tmppdf/.
cd tmppdf

# compter les pages
N=$(pdftk book.pdf dump_data | grep NumberOfPages | cut -d: -f2)

# si besoin, ajouter une page pour avoir un nombre paire
if [ $N -ne $[($N)/4*2] ]
then
        pdftk A=book.pdf cat A1-end A1 output book2.pdf
        mv book2.pdf book.pdf
fi

# millieu du document
N=$[($N+1)/2]

# extraire les future pages:
# recto: N / 1
# verso: 2 / N-1
pdftk A=book.pdf cat A1-${N}odd output out2.pdf
pdftk A=book.pdf cat A1-${N}evenS output out4.pdf
pdftk A=book.pdf cat Aend-$[N+1]even output out1.pdf
pdftk A=book.pdf cat Aend-$[N+1]oddS output out3.pdf

# éclater les 4 pdf ci-dessus => pages dans l'ordre alphabétique
pdftk out1.pdf burst output %05d_out1.pdf
pdftk out2.pdf burst output %05d_out2.pdf
pdftk out3.pdf burst output %05d_out3.pdf
pdftk out4.pdf burst output %05d_out4.pdf

# fusioner toutes les pages obtenues ci-dessus
pdftk 00*pdf cat output merge.pdf

# A4 => 2xA5
pdfnup --nup 2x1 merge.pdf --orient landscape --outfile miniBook.pdf

cp miniBook.pdf ..
```

- crypt/decrypt

```
encrypt() {
        gpg -c $1 && rm $1
}
decrypt() {
        [ ${1%.gpg} != $1 ] || return
        gpg -d $1 > ${1%.gpg} 2>/dev/null && rm $1
}
```

- renomer un fichier sans les accents

```
#!/bin/bash

eAcute=$(echo "é" | iconv -f UTF-8 -t ISO-8859-1)
eGrave=$(echo "è" | iconv -f UTF-8 -t ISO-8859-1)
eChapo=$(echo "ê" | iconv -f UTF-8 -t ISO-8859-1)
eTrema=$(echo "ë" | iconv -f UTF-8 -t ISO-8859-1)
aGrave=$(echo "à" | iconv -f UTF-8 -t ISO-8859-1)
aChapo=$(echo "â" | iconv -f UTF-8 -t ISO-8859-1)

SRC="${eAcute}${eGrave}${eChapo}${eTrema}${aGrave}${aChapo} '\""
DST="eeeeaa___"

for f1 in $1/*; do
    # pour ne pas avoir de caractères codés sur plusieurs octets
    # (sinon `tr "é" "e"` remplace "é" par "ee")
    f2=$(echo $f1 | iconv -f UTF-8 -t ISO-8859-1)

    # remplacement des caractères spéciaux
    f3=$(echo $f2 | tr "$SRC" "$DST")

    # renomage
    echo mv \"$f1\" $f3
done
```

- générer un mot de pase pour */etc/shadow*

```
// cf man shadow
// cf man crypt
// gcc -Wall -o crypt crypt.c -lcrypt

#define _XOPEN_SOURCE       /* See feature_test_macros(7) */

#include <stdio.h>
#include <unistd.h>

int main(int argc, char** argv)
{
char* salt = 0;
char* key = 0;

if (argc < 3) {
printf("usage: %s 'salt' 'passwd_en_clair'\n", argv[0]);
```

```
printf(" avec salt: '$[1-6]$...'\n");
return 1;
}

salt = argv[1];
key =  argv[2];

printf("salt='%s', clair='%s'\n", salt, key);
printf("crypte='%s'\n", crypt(key, salt));

return 0;
}
```