??? progress bar

???

```
echo -ne '>>>                       [20%]\r'
# some task
sleep 2
echo -ne '>>>>>>>                   [40%]\r'
# some task
sleep 2
echo -ne '>>>>>>>>>>>>>>             [60%]\r'
# some task
sleep 2
echo -ne '>>>>>>>>>>>>>>>>>>>>>>>   [80%]\r'
# some task
sleep 2
echo -ne '>>>>>>>>>>>>>>>>>>>>>>>>>>[100%]\r'
echo -ne '\n'
```

?????1

```
#!/bin/bash
#
# inprogress.sh
# Description: this script may show some simple motion in foreground
# when main program in progress.
# Author: A-Lang (alang[dot]hsu[at]gmail[dot]com)

trap "kill $BG_PID;echo;exit" 1 2 3 15

function waiting
{
    INTERVAL=0.5
    TT='->——
    —>——
    —>—-
```

```
    –->–
    ——>–
    ——>-
    ——->
    ——<-
    ——<–
    –-<–
    –<–-
    –<——
    -<——
    <——-';
    echo -ne "Please waiting… \n"
    while true
    do
        for j in $TT
        do
            echo -ne "Please be patient, $j\r"
            sleep $INTERVAL
        done

    done
}

# Start the waiting, where place your main program.
waiting &
BG_PID=$!

sleep 10

# End of the waiting
kill $BG_PID
echo
```

?????2

- YT: [Creating an Animated Progress Bar from scratch in Bash to batch-process files! - YouTube]

```
#!/usr/bin/env bash

BATCHSIZE=1

progress-bar() {
  local current=$1
  local len=$2

  local bar_char='|'
  local empty_char=' '
  local length=50
  local perc_done=$((current * 100 / len))
  local num_bars=$((perc_done * length / 100))

  local i
  local s='['
  for ((i = 0; i < num_bars; i++)); do
    s+=$bar_char
  done
  for ((i = num_bars; i < length; i++)); do
    s+=$empty_char
  done
  s+=']'

  echo -ne "\r$s $current/$len ($perc_done%)"
}

process-files() {
  local files=("$@")

  sleep .01
}

shopt -s globstar nullglob

echo 'finding files'
files=(./**/*cache)
len=${#files[@]}
echo "found $len files"
```

```
for ((i = 0; i < len; i += BATCHSIZE)); do
progress-bar "$((i+1))" "$len"
process-files "${files[@]:i:BATCHSIZE}"
done
progress-bar "$len" "$len"

echo
```

????

```
#!/usr/bin/env bash

spinner()
{
    local pid=$!
    local delay=${1:1}
    local spinstr='|/-\'
    while [ "$(ps a | awk '{print $1}' | grep $pid)" ]
    do
        local temp=${spinstr#?}
        printf " [%c] " "$spinstr"
        local spinstr=$temp${spinstr%"$temp"}
        sleep $delay
        printf "\b\b\b\b\b"
    done
    printf "    \b\b\b\b"
}

# Run your program here
#(a_long_running_task) &
sleep 10 &
spinner 0.75
```

?? LED ???

```
#!/bin/bash

echo "Starting to format the disk...."
```

```
# Start to blink the LED
/root/bin/led.sh b
/root/bin/led.sh blink &
led_pid=$!

# Main codes
echo "formating hdisk2..."
sleep 10

# When the main codes is ending up
# Stop to blink the LED
kill -9 $led_pid
wait $led_pid 2>/dev/null
sleep 2

# Making sure the LED is ON with Blue
/root/bin/led.sh b
echo "done"
```

> led.sh ???? LED ???
>
> wait .... ?????? killed ... ?????

Revision #11
Created 2020-10-27 12:47:46 CST by A-Lang (Admin)
Updated 2025-09-06 09:31:15 CST by A-Lang (Admin)