LogicOS 2026

Pourquoi LogicOS ?
LogicOS équipe les ordinateurs mis à disposition des adhérent·e·s de l'association. On peut y retrouver les logiciels spécifiques utilisés avec les machines de fabrication numérique du fablab, en particulier inkscape/inkstitch, l'extension inkscape-silhouette pour piloter la Cameo 4, cura slicer configuré pour les imprimantes 3D, etc.
Jusque là, nous avions des ordis installés au fil du temps avec des installations de logiciels différentes et différentes versions. Bref, le bazar!

En 2026, il s'agit de la première version, qui ne sera pas exempte de bugs et nécessitera de futures améliorations. Il est possible d'indiquer en bas de page les bugs rencontrés et les améliorations souhaitées.

Quelle système ?
LogicOS est basé sur **Debian 13** avec l'environnement de bureau **Gnome**

Utilisation
Utilisateur : lesporteslogiques
Mot de passe : lpl
Ce compte ne peut pas utiliser sudo. La connexion en tant que superutilisateur est accessible à l'utilisateur root avec un mot de passe spécifique

Mise en place
Les logiciels ont été installés et configurés sur un premier ordinateur par qui a permis de créer un *master*, puis clonés avec clonezilla. La procédure est documentée sur les pages suivantes :

- création du master et clonage
- éléments de configuration

La version de LogicOS 2026 a été réalisée par Jejy dans le cadre d'un stage de BTS SIO.

Logiciels installés

Liste de tous les logiciels avec la version installée pour Logic OS 2026.
Si le nom du logiciel est cliquable, le lien renvoie vers la page dédiée de ce wiki.
Origine renvoie au site officiel du logiciel.

Logiciels système

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
terminator		2.1.4	apt
etcher	origine	2.1.4	Paquet .deb (Github)
gsconnect		62	apt
syncthing	origine	1.29.5	apt
git		2.47.3	apt
imhex	origine	1.37.4	appimage
pdfjam	origine	4.1.0	apt
poppler-utils	origine	3.03	apt
pdfposter		0.8.1	apt

Notes

- Etcher : https://lesporteslogiques.net/wiki/materiel/installation_ordinateurs#etcher_-_installation_sous_debian_13
- GSconnect : en remplacement de kdeconnect (meilleure intégration avec GNOME)
- syncthing : créer ~/.local/state/syncthing/syncthing.lock si absent

Logiciels de traitement de médias numériques en ligne de commande

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
ffmpeg		7.1.2	apt
imagemagick		7.1.1	apt
sox		14.4.2	apt

Notes

- Sox : installer le paquet “libsox-fmt-all” pour la gestion des formats mp3 et ogg

Logiciels de création sonore

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
audacity		3.7.3	apt
reaper	origine		Archive
pure data		0.55.2	apt
plugdata	origine	0.9.2	apt
automatonism	origine	3.1.0	Archive site
sonic pi		4.6.0	Github
chuck			apt
~~miniaudicle~~			
troop / foxdot			Archive (Github)
audacious		4.4.2	apt
supercollider IDE		3.13.0	apt
mixxx		2.5.0	apt
QSynth		2.4.4	apt
soundthread		0.4.0	github
furnace		0.6.8.1	apt
orca		48.1	apt
pilot		2.99	apt
liquidsoap	origine	2.3.2	apt
VCV rack			archive site
jack (server)		1.9.22	apt

Notes

- Reaper, régler Preferences - Audio - Device - Audio system : PulseAudio
- miniaudicle : erreurs de compilation non résolues
- VCV Rack : raccourci sur le breau non fonctionnel

Logiciels de création graphique

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
gimp		3.0.4	apt
inkscape		1.4.3	AppImage
inkstich		3.2.2	Script sh
extension inkscape silhouette		1.29-1	Github
scribus		1.6.3	apt
fontforge		20230101	apt
krita		5.2.9	apt

Notes

- extension inkscape-silhouette : dépendance python3-usb nécessaire + python3-matplotlib + python3-tinycss2
- probleme du bug qui ferme inkscape quand on ouvre le menu préferences : effacer le dossier '~/.local/share/icc' (certains profils utilisent des caractères ISO-8859-1 alors que l'encodage local est en UTF-8 : source)

Logiciels de print

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
ql570			
Stickeuse QL570			

Guide d'utilisation de StickeuseQL570

Logiciels de vidéo / création vidéo

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
VLC		3.0.21	apt
kdenlive		24.12.3	apt
cheese		44.1	apt
OBS Studio		30.2.3.1-3	apt
Friction	origine	0.9.6.1	AppImage
Jubler		9.0.1	AppImage
aegisub			

Logiciels pour l'électronique

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
kicad		v9.0	
arduino		1.8.19	
arduino > 2	origine	2.3.6	
fritzing	origine	1.0.1	apt

Notes

- 2 versions d'Arduino installées en parallèle

Logiciels pour la fabrication numérique

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
freecad		1.0	apt
openscad		2021.01	apt
cura slicer		5.0.0 et 5.11.0	apt

Logiciels de création 3D

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
blender		4.3.2	apt
gmsh		4.13.1	apt

Code & code créatif

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
vscodium	origine	1.106.27818	
processing	origine	4.4.10	Snap
notepadqq		2.0.0	apt
python3 + pipx			apt
vpype		1.15.0	pipx
~~ln~~			

Notes

- python 3.13 installé

Logiciels de création de jeux vidéo

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
GBStudio		4.1.3	AppImage
LibreSprite		1.1	AppImage
Tiled		1.11.2	AppImage
Bitsy local		7.2	Raccourci HTML

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
Ren'Py		8.3.4	apt
godot		4.5.1	Exécutable site
twine		2.11.1	Archive Github

Divers

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
libreoffice		25.2.3.2	
logseq		0.10.15	AppImage

Logiciels pour le tricot machine

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
knittington			

Notes

- Knittington nécessite d'être compilé en fonction des dépendances actuelles nécessaires...

Éléments de configuration spécifique

Configuration du moteur de recherche du wiki dans firefox : utiliser @wikilpl dans la barre de recherche pour utiliser le moteur de recherche interne du wiki

Aide à l'accessibilité

Nom (et lien vers la page de ce wiki)	Origine (lien projet)	Version	Type d'installation
gnome-shell-extensions			apt
Keyboard Modifiers Status	origine		extension gnome

Notes

- On trouve gnome-shell-extensions en cherchant «gestionnaire d'extensions»
- gnome-shell-extensions s'installe sous le nom «extenqsion-manager»

Comment cette liste a été réalisée?

Cette liste a été extraite au format csv depuis le framacalc qui a servi à l'établir puis transformée au format dokuwiki par un script python écrit par un LLM, puis corrigée manuellement dans dokuwiki! Le script est ci-dessous.

csv_to_dokuwiki.py (cliquer pour afficher le code)

csv_to_dokuwiki.py

```
#!/usr/bin/env python3

"""
utilisation : python3 csv_to_dokuwiki.py systeme-logicos26.csv > sortie.txt

20260316 / python 3.1.2 @ Tenko / code python réalisé par un LLM

prompt :

Script python3, sans dépendances pour passer d'un fichier csv à un format tableau dokuwiki.
Voila comment traiter chaque colonne :
la ligne 1 fournit les entêtes
la ligne 2 n'est pas à traiter
Pour toutes les lignes suivantes
colonne A : si rempli, créer un nouveau tableau avec les lignes qui suivent
colonne B : ne pas traiter
colonne C : traiter uniquement si le caractère ! apparaît dans ce cas il faudra barrer le nom de la colonne D
colonne D : faire apparaître le nom, si la colonne E est remplie faire de ce nom un lien avec l'url de la colonne E colonne E :
voir colonne D
colonne F : afficher «origine» et créer un lien vers l'URL contenue dans la colonne F
colonne G : afficher
colonne H : afficher
colonne I : reprendre le texte en bas de page hors des tableaux
colonne J : ne pas traiter
colonne K : reprendre le texte en bas de page hors des tableaux
colonne L : reprendre en bas de page
```

```
"""

import csv
import sys

def dokuwiki_link(text, url):
    if url:
        return f"[[{url}|{text}]]"
    return text

def strike(text):
    return f"<del>{text}</del>"

def main(csv_file):
    footnotes = []

    with open(csv_file, newline='', encoding="utf-8") as f:
        reader = csv.reader(f)

        rows = list(reader)

    headers = rows[0]
    current_table_open = False

    for i, row in enumerate(rows[2:], start=3):  # ligne 3 et suivantes

        colA = row[0].strip() if len(row) > 0 else ""
        colC = row[2].strip() if len(row) > 2 else ""
        colD = row[3].strip() if len(row) > 3 else ""
        colE = row[4].strip() if len(row) > 4 else ""
        colF = row[5].strip() if len(row) > 5 else ""
        colG = row[6].strip() if len(row) > 6 else ""
        colH = row[7].strip() if len(row) > 7 else ""
        colI = row[8].strip() if len(row) > 8 else ""
        colK = row[10].strip() if len(row) > 10 else ""
        colL = row[11].strip() if len(row) > 11 else ""

        # nouveau tableau si colonne A remplie
        if colA:
            if current_table_open:
                print()
            print(f"===== {colA} =====")
            print("^ Nom ^ Origine ^ G ^ H ^")
            current_table_open = True

        # traitement colonne D (nom + lien éventuel)
        name = colD
        if colE:
            name = dokuwiki_link(name, colE)

        # si ! dans colonne C → barrer
        if "!" in colC:
            name = strike(name)

        # colonne F → lien origine
        origine = ""
        if colF:
            origine = f"[[{colF}|origine]]"

        print(f"| {name} | {origine} | {colG} | {colH} |")

        # collecte bas de page
        for note in (colI, colK, colL):
            if note:
                footnotes.append(note)

    # affichage notes hors tableau
    if footnotes:
        print("\n----\n")
        for n in footnotes:
            print(f"* {n}")

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: csv_to_dokuwiki.py fichier.csv")
        sys.exit(1)

    main(sys.argv[1])
```

Bugs rencontrés

Améliorations souhaitées pour les prochaines versions

- Installation de MPlayer (nécessaire pour Jubler, merci)
- Ajouter Anaouder

Désinstaller l'ancienne Stickeuse QL-570 des OP[XX] (image de janvier 2026)

Protocole établi le 29 juillet 2026 — script version 1.1.

Périmètre — les ordinateurs portables **OP[XX]** des Portes Logiques.

Qui l'applique — Pierre, ou toute personne qu'il habilite. L'opération exige les droits root et intervient dans les dossiers personnels de **tous** les comptes du poste : elle ne s'improvise pas.

Les OP[XX] ont été déployés par **clonage**, par Jérôme (Jejy), en décembre 2025 / janvier 2026. L'**ancienne** version de la Stickeuse figurait dans l'image clonée : elle est donc présente sur tous les postes issus de ce déploiement, **aux mêmes emplacements** — `/opt/stickeuse_ql570`, `~/bin/stickeuse_ql570`, `~/.gitdl/ql570`, l'icône et les deux lanceurs. Elle n'a pas de désinstalleur fiable.

Un poste peut s'écarter de ce motif s'il a reçu des installations manuelles ultérieures : OP42, poste de développement, porte en plus une règle udev, une entrée d'autostart et un binaire dans `/usr/local/bin`. Le script couvre les deux cas.

Il faut la retirer **avant** d'installer la version actuelle, sinon les deux se marchent dessus : deux entrées de menu portant le même nom, un agent fantôme relancé à chaque ouverture de session, une règle udev obsolète.

Ordre : **nettoyer, puis installer**.

- Version actuelle de l'appli : dépôt lesporteslogiques/stickeuse

Utilisation

Télécharger le script ci-dessous, puis, sur le poste à nettoyer :

```
su -
cd /chemin/vers/le/script
bash nettoyage_stickeuse.sh -n     # inventaire seul, ne supprime rien
bash nettoyage_stickeuse.sh        # supprime, après confirmation
```

Pas besoin de `chmod +x` : lancer par `bash nom_du_script` suffit.

Option	Effet
`-n`, `--inventaire-seul`	Affiche ce qui serait supprimé, puis s'arrête.
`-y`, `--oui`	Ne demande pas confirmation. Pour enchaîner les postes.
`--avec-pip`	Désinstalle aussi la bibliothèque Python `brother_ql`.
`-h`, `--aide`	Affiche l'aide intégrée.

Ce qu'il fait

1. **Inventaire** — cherche les traces sans rien supprimer, côté système (`/opt/stickeuse_ql570`, `/usr/local/bin/ql570`, lanceurs, règles udev) et dans chaque compte de `/home` (`bin/stickeuse_ql570`, `.gitdl/ql570`, `.icons`, `.config`, `.cache`, `.local/share`, `autostart`, `Bureau`, `Desktop`). Aucun nom de compte n'est écrit en dur.
2. **Confirmation** — affiche la liste et attend un o. Toute autre réponse annule.
3. **Suppression** — efface, recharge les règles udev, rafraîchit le cache des menus.
4. **Vérification** — relance la recherche et annonce « Aucun reste » ou la liste de ce qui subsiste.

Chaque passage écrit un journal `nettoyage_<hôte>_<horodatage>.log` dans le dossier courant. Lancé depuis une clé USB, les journaux s'y accumulent : c'est le suivi du parc.

Garde-fou : ancienne ou nouvelle version ?

Le script s'appuie sur la ligne Exec= des fichiers `.desktop`, qui indique le programme réellement lancé :

Exec= pointe vers	Version	Décision
`/opt/stickeuse_ql570/…` ou `/usr/local/bin/ql570`	ancienne	supprimé
`/opt/ql570/venv/bin/python …`	nouvelle	conservé et signalé

La nouvelle version n'est **jamais** supprimée.

Le cas brother_ql

`brother_ql` est la bibliothèque Python qui pilote les imprimantes QL. Installée par `pip` en root dans `/usr/local`, elle est **partagée par tout le poste**. Elle n'appartient pas à l'ancienne version — celle-ci passait par un binaire C compilé.

Sur OP60, elle est datée du **8 janvier 2026**, soit dans la fenêtre du clonage : elle vient donc probablement de l'image, et non d'un usage personnel. Si la même date se retrouve sur les autres postes, l'hypothèse se confirme et `--avec-pip` devient le passage par défaut. Par prudence, le script la **conserve** tant que la question n'est pas tranchée, et signale seulement sa présence.

Marche à suivre sur le parc

Les postes étant issus d'un même clonage, l'inventaire devrait être identique de l'un à l'autre. C'est ce qui rend le passage en série possible — et ce qu'il faut vérifier d'abord.

1. Sur les trois ou quatre premiers postes, passer en `-n` et lire l'inventaire. Il doit être le même partout. Un écart signale une installation manuelle ultérieure : la relever avant de continuer.
2. Une fois le motif confirmé, enchaîner les postes suivants.
3. Noter la date de `brother_ql` sur ces premiers postes (voir plus haut).
4. Récupérer les journaux pour tenir le décompte du parc.

Limites connues

- **Une seule partition** — la recherche utilise `-xdev` et ne franchit pas les frontières de systèmes de fichiers. Si `/home` est sur une partition séparée (`df -h / /home /opt` pour le vérifier), relancer à la main : `find /home -iname '*stickeuse*' -o -iname '*ql570*' .`
- **Emplacements connus seulement** — le script cherche dans les dossiers observés sur deux postes (OP42, poste de développement, et OP60, issu du clonage). Une installation posée ailleurs lui échapperait, d'où le passage en `-n` sur les premiers postes.
- **Fermer la session** — l'affichage du menu ne se rafraîchit complètement qu'à la réouverture de session.
- **Corbeille** — les fichiers déjà mis à la corbeille par un utilisateur (`~/.local/share/Trash`) ne sont pas concernés.

Récupérer le script

Le script est publié en bas de cette page. Le bloc de code porte son nom en en-tête : **cliquer dessus télécharge le fichier**. Préférer ce lien au copier-coller, qui risque d'emporter des espaces ou de tronquer une ligne.

Deux façons de l'amener sur les postes :

- **Depuis le poste lui-même** — ouvrir cette page dans le navigateur du poste, télécharger, puis lancer depuis `~/Téléchargements`.
- **Par clé USB** — télécharger une fois, copier sur la clé, passer de poste en poste. À privilégier : les journaux s'accumulent sur la clé et donnent le suivi du parc.

Contrôle après téléchargement, avant la première utilisation :

```
wc -l nettoyage_stickeuse.sh     # doit afficher 334 lignes
bash -n nettoyage_stickeuse.sh   # vérifie la syntaxe, n'exécute rien
```

Si la seconde commande ne dit rien, le fichier est intact.

Le script

Auteurice : Vitally LUBIN — Les Portes Logiques, 2026. Licence : GNU Affero General Public License v3 (AGPL-3.0), comme le reste du projet Stickeuse — et non la licence CC BY-SA du wiki (le pied de page réserve ce cas : *sauf mention contraire*).

nettoyage_stickeuse.sh

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

```bash
#  nettoyage_stickeuse.sh  –  version 1.1
#
#  Retire l'ANCIENNE Stickeuse QL-570 d'un poste OP (Les Portes Logiques).
#
#  Auteurice : Vitally LUBIN – Les Portes Logiques, 2026
#  Licence   : GNU Affero General Public License v3 (AGPL-3.0)
#              https://www.gnu.org/licenses/agpl-3.0.html
#
#  POURQUOI
#      Faire place nette AVANT d'installer la version aboutie du dépôt :
#      https://github.com/lesporteslogiques/stickeuse
#      Sans ce ménage, les résidus de l'ancienne version (lanceurs orphelins,
#      agent fantôme, règle udev obsolète) parasitent le déploiement.
#      Ordre : nettoyer, puis installer.
#
#  UTILISATION RAPIDE
#      su -
#      bash nettoyage_stickeuse.sh -n     # inventaire seul, ne supprime rien
#      bash nettoyage_stickeuse.sh        # inventorie, montre, demande, supprime
#
#  OPTIONS
#      -n, --inventaire-seul   affiche ce qui serait supprimé, puis s'arrête
#      -y, --oui               ne demande pas confirmation (passage en série)
#          --avec-pip          désinstalle aussi la bibliothèque Python brother_ql
#      -h, --aide              affiche cette aide
#
#  DÉROULÉ
#      1. Inventaire   : cherche les traces, ne touche à rien
#      2. Confirmation : affiche la liste, attend un « o »
#      3. Suppression  : efface, recharge udev, rafraîchit les menus
#      4. Vérification : relance la recherche et annonce ce qui subsiste
#
#  GARDE-FOU
#      La NOUVELLE version (/opt/ql570 et son venv) n'est JAMAIS supprimée.
#      Tout lanceur .desktop pointant vers elle est conservé et signalé.
#
#  Documentation complète : voir README.md
# ==============================================================================

# set -u : une variable non définie devient une erreur au lieu d'un vide
#          silencieux. Sur un script qui fait des rm, c'est vital : « rm -rf
#          $CHEMIN » avec CHEMIN vide viserait la racine.
# set -o pipefail : dans « a | b », le code retour est celui du premier
#          maillon qui échoue, pas seulement du dernier.
# (pas de « set -e » : on veut continuer le nettoyage même si un rm échoue,
#  et le signaler dans le journal.)
set -uo pipefail

VERSION="1.1"
HOTE="$(hostname)"                        # sert à nommer le journal par poste
HORODATE="$(date +%Y%m%d-%H%M%S)"
JOURNAL="$(pwd)/nettoyage_${HOTE}_${HORODATE}.log"

# Drapeaux, modifiés plus bas par les options de la ligne de commande
INVENTAIRE_SEUL=0
SANS_CONFIRMATION=0
AVEC_PIP=0

# Liste blanche : chemins de la NOUVELLE version, jamais supprimés.
# Ajouter ici tout autre chemin à préserver.
PROTEGES=("/opt/ql570")

CIBLES=()        # ce qui sera supprimé
SIGNALES=()      # ce qui est conservé volontairement, mais mérite d'être dit

# ------------------------------------------------------------------ utilitaires

# Tout ce qui s'affiche part aussi dans le journal : « tee -a » écrit sur la
# sortie standard ET ajoute au fichier. Sur 30 postes, le journal est la seule
# trace de ce qui a été trouvé et supprimé sur chacun.
msg()   { printf '%s\n' "$*" | tee -a "$JOURNAL"; }
titre() { printf '\n=== %s ===\n' "$*" | tee -a "$JOURNAL"; }

# Le chemin appartient-il à la nouvelle version ?
# Le motif "$p"/* attrape aussi tout ce qui est DANS un dossier protégé.
est_protege() {
  local chemin="$1" p
  for p in "${PROTEGES[@]}"; do
    [[ "$chemin" == "$p" || "$chemin" == "$p"/* ]] && return 0
  done
  return 1
}

# Ajoute un chemin à la liste de suppression, avec trois filtres :
#   1. chemin vide      -> ignoré (protection anti-« rm -rf » sur rien)
#   2. n'existe pas     -> ignoré (inutile d'encombrer la liste)
#   3. protégé          -> ignoré (nouvelle version)
#   4. déjà présent     -> ignoré (évite les doublons)
ajouter() {
  local chemin="$1"
  [[ -z "$chemin" ]] && return
  [[ -e "$chemin" ]] || return
  est_protege "$chemin" && return
  local c
```

```
  # ${CIBLES[@]+"${CIBLES[@]}"} : formule qui évite l'erreur « variable non
  # définie » (set -u) quand le tableau est encore vide.
  for c in ${CIBLES[@]+"${CIBLES[@]}"}; do
    [[ "$c" == "$chemin" ]] && return
  done
  CIBLES+=("$chemin")
}

signaler() { SIGNALES+=("$1"); }

# L'aide est l'en-tête du fichier lui-même : une seule source, jamais de
# décalage entre la doc et le code.
aide() { sed -n '4,38p' "$0" | sed 's/^# \{0,1\}//'; exit 0; }

# ------------------------------------------------------------------ arguments

while [[ $# -gt 0 ]]; do
  case "$1" in
    -n|--inventaire-seul) INVENTAIRE_SEUL=1 ;;
    -y|--oui)             SANS_CONFIRMATION=1 ;;
    --avec-pip)           AVEC_PIP=1 ;;
    -h|--aide)            aide ;;
    *) echo "Option inconnue : $1 (voir --aide)" >&2; exit 2 ;;
  esac
  shift
done

# EUID = identifiant effectif de l'utilisateur ; 0 = root.
# Nécessaire pour /opt et /etc, et pour parcourir tous les /home.
if [[ "${EUID}" -ne 0 ]]; then
  echo "Ce script doit tourner en root : su - puis bash $0" >&2
  exit 1
fi

msg "nettoyage_stickeuse.sh v${VERSION} — poste ${HOTE} — ${HORODATE}"
msg "Journal : ${JOURNAL}"

# ============================================================ 1. INVENTAIRE
# Rien n'est supprimé ici : on ne fait que remplir CIBLES et SIGNALES.

titre "1. Inventaire"

# --- côté système ------------------------------------------------------------
ajouter "/opt/stickeuse_ql570"      # dossier de l'ancienne version
ajouter "/usr/local/bin/ql570"      # binaire C compilé de l'ancienne version

# Lanceurs du menu général. C'est ici que se joue la distinction ancienne /
# nouvelle : la ligne « Exec= » du fichier .desktop dit quel programme est
# lancé. Si elle pointe vers /opt/ql570, c'est la NOUVELLE -> on conserve.
for f in /usr/share/applications/*ql570* /usr/share/applications/*stickeuse*; do
  [[ -e "$f" ]] || continue         # le motif non résolu reste littéral : on saute
  if grep -qs 'Exec=.*/opt/ql570' "$f"; then
    signaler "$f (lanceur de la NOUVELLE version — conservé)"
  else
    ajouter "$f"
  fi
done

# Règles udev : elles donnent à l'utilisateur le droit de parler à
# l'imprimante USB. Absentes sur OP60, présentes sur OP42.
for f in /etc/udev/rules.d/*ql570* /etc/udev/rules.d/*brother-ql*; do
  [[ -e "$f" ]] && ajouter "$f"
done

# --- côté utilisateurs -------------------------------------------------------
# On balaie /home/* plutôt que d'écrire un nom de compte en dur : il vaut
# « xor » sur OP42, « lesporteslogiques » sur OP60, et sans doute autre chose
# ailleurs. Le script s'adapte sans être modifié.
for maison in /home/*; do
  [[ -d "$maison" ]] || continue

  ajouter "$maison/bin/stickeuse_ql570"   # dossier de travail (OP60)
  ajouter "$maison/.gitdl/ql570"          # sources téléchargées (OP60)
  ajouter "$maison/.ql570"                # journal d'impression de l'appli

  # Recherche par motif dans les dossiers de configuration.
  # -maxdepth 1 : on ne descend pas dans les sous-dossiers, pour ne pas
  # ramasser un fichier sans rapport enfoui quelque part.
  for d in "$maison/.icons" "$maison/.config" "$maison/.cache" \
           "$maison/.local/share" "$maison/.config/autostart"; do
    [[ -d "$d" ]] || continue
    # « while read < <(commande) » : on lit ligne par ligne sans sous-shell,
    # sinon les ajouts au tableau CIBLES seraient perdus.
    while IFS= read -r f; do
      ajouter "$f"
    done < <(find "$d" -maxdepth 1 \( -iname '*stickeuse*' -o -iname '*ql570*' \) 2>/dev/null)
  done

  # Lanceurs personnels : menu utilisateur, Bureau (session en français) et
  # Desktop (session en anglais). Même test Exec= que plus haut.
  for d in "$maison/.local/share/applications" "$maison/Bureau" "$maison/Desktop"; do
    [[ -d "$d" ]] || continue
    while IFS= read -r f; do
      if grep -qs 'Exec=.*/opt/ql570' "$f"; then
```

```
        signaler "$f (lanceur de la NOUVELLE version — conservé)"
      else
        ajouter "$f"
      fi
    done < <(find "$d" -maxdepth 1 \( -iname '*stickeuse*' -o -iname '*ql570*' \) 2>/dev/null)
  done
done

# --- bibliothèque Python brother_ql ----------------------------------------
# Installée par pip en root dans /usr/local, donc partagée par tout le poste.
# Elle n'appartient PAS à l'ancienne version (qui utilisait un binaire C) :
# quelqu'un a pu l'installer pour son propre usage. Conservée par défaut.
PIP_PRESENT=0
if command -v brother_ql >/dev/null 2>&1 || \
   ls -d /usr/local/lib/python3.*/dist-packages/brother_ql >/dev/null 2>&1; then
  PIP_PRESENT=1
fi

# --- restitution de l'inventaire -------------------------------------------
if [[ ${#CIBLES[@]} -eq 0 ]]; then
  msg "Aucune trace de l'ancienne Stickeuse sur ce poste."
else
  msg "${#CIBLES[@]} élément(s) à supprimer :"
  for c in "${CIBLES[@]}"; do msg "  - $c"; done
fi

if [[ ${#SIGNALES[@]} -gt 0 ]]; then
  msg ""
  msg "Conservé (non concerné par l'ancienne version) :"
  for s in "${SIGNALES[@]}"; do msg "  ! $s"; done
fi

if [[ $PIP_PRESENT -eq 1 ]]; then
  msg ""
  if [[ $AVEC_PIP -eq 1 ]]; then
    msg "brother_ql (pip) : présent — sera désinstallé (--avec-pip)."
  else
    msg "brother_ql (pip) : présent — CONSERVÉ. Relancer avec --avec-pip pour l'ôter."
  fi
fi

if [[ $INVENTAIRE_SEUL -eq 1 ]]; then
  msg ""
  msg "Mode inventaire seul : rien n'a été supprimé."
  exit 0
fi

# Poste déjà propre : on s'arrête là.
[[ ${#CIBLES[@]} -eq 0 && $PIP_PRESENT -eq 0 ]] && exit 0

# ============================================================ 2. CONFIRMATION

if [[ $SANS_CONFIRMATION -eq 0 ]]; then
  printf '\nSupprimer les éléments ci-dessus ? [o/N] '
  read -r reponse
  echo "Réponse : ${reponse:-N}" >> "$JOURNAL"
  case "${reponse}" in
    [oOyY]) ;;                                  # o, O, y ou Y : on continue
    *) msg "Abandon : rien n'a été supprimé."; exit 0 ;;   # tout le reste : on sort
  esac
fi

# ============================================================ 3. SUPPRESSION

titre "2. Suppression"

UDEV_TOUCHE=0
for c in ${CIBLES[@]+"${CIBLES[@]}"}; do
  est_protege "$c" && continue        # second contrôle, au cas où la liste
                                      # blanche aurait évolué entre-temps
  [[ "$c" == /etc/udev/rules.d/* ]] && UDEV_TOUCHE=1
  # « -- » marque la fin des options : un fichier nommé « -rf » serait traité
  # comme un nom, pas comme un drapeau.
  if rm -rf -- "$c"; then
    msg "  supprimé : $c"
  else
    msg "  ÉCHEC    : $c"
  fi
done

# Sans rechargement, une règle supprimée reste active jusqu'au redémarrage.
if [[ $UDEV_TOUCHE -eq 1 ]]; then
  udevadm control --reload-rules 2>/dev/null && msg "  règles udev rechargées"
fi

# Debian 13 refuse pip en root sans --break-system-packages (PEP 668).
# Sans risque ici : ces fichiers vivent dans /usr/local, hors du Python géré
# par apt. Le « || » relance sans le drapeau sur les Debian plus anciennes.
if [[ $PIP_PRESENT -eq 1 && $AVEC_PIP -eq 1 ]]; then
  msg "  désinstallation de brother_ql (pip)"
  pip3 uninstall -y --break-system-packages brother_ql 2>&1 | tee -a "$JOURNAL" \
    || pip3 uninstall -y brother_ql 2>&1 | tee -a "$JOURNAL"
fi
```

```
# Le menu lit un cache : sans ce rafraîchissement, l'entrée reste affichée
# alors que le fichier a disparu (clic sans effet).
if command -v update-desktop-database >/dev/null 2>&1; then
  update-desktop-database /usr/share/applications 2>/dev/null
  for maison in /home/*; do
    [[ -d "$maison/.local/share/applications" ]] && \
      update-desktop-database "$maison/.local/share/applications" 2>/dev/null
  done
  msg "  menus rafraîchis"
fi

# ============================================================ 4. VERIFICATION
# On relance la recherche à blanc sur tout le disque pour confirmer.

titre "3. Vérification"

# -xdev : ne franchit pas les frontières de systèmes de fichiers. Évite de
# partir dans /proc et /sys. Si /home est sur une autre partition, il faut
# relancer la recherche dessus (voir README, section « Limites »).
#
# On écarte de la liste des « restes » : la nouvelle version, ce qui a été
# volontairement conservé, le script lui-même et son journal — sinon le script
# signalerait comme suspect ce qu'il vient de décider de garder.
MOI="$(readlink -f "$0" 2>/dev/null || echo "$0")"
RESTES="$(find / -xdev \( -iname '*stickeuse*' -o -iname '*ql570*' \) 2>/dev/null \
          | grep -v '^/opt/ql570' \
          | grep -Fxv "$MOI" \
          | grep -Fv "$(basename "$MOI")" \
          | grep -Fxv "$JOURNAL" || true)"

for s in ${SIGNALES[@]+"${SIGNALES[@]}"}; do
  chemin="${s%% (*}"                  # retire le commentaire entre parenthèses
  RESTES="$(printf '%s\n' "$RESTES" | grep -Fxv "$chemin" || true)"
done

if [[ -z "$RESTES" ]]; then
  msg "Aucun reste. Poste ${HOTE} propre."
else
  msg "Restes à examiner :"
  printf '%s\n' "$RESTES" | tee -a "$JOURNAL"
fi

if command -v brother_ql >/dev/null 2>&1; then
  msg "brother_ql (pip) toujours en place (conservé volontairement, ou --avec-pip non passé)."
fi

msg ""
msg "Terminé. Fermer et rouvrir la session pour rafraîchir l'affichage du menu."
msg "Journal : ${JOURNAL}"
```

Article extrait de : http://www.lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://www.lesporteslogiques.net/wiki/materiel/logicos/start?rev=1785333754**
Article mis à jour: **2026/07/29 16:02**