# \$> S-0S

L'OS qui résout tous vos problèmes en créant de nouveaux problèmes.

# \$> USERS

- + Guigui
- + Paulux
- + AnTiToinE
- + Huggy
- + Ferrero
- + Estounette
- + Lisouille

#### \$> Fonctionnalités

État gestion mémoire : + mmu...ok. + allocation dynamique...ok. + isolation...ok. Autres fonctionnalités : + sortie écran...ok. + entrée clavier...ok. Ordonnancement: ordonnancement collaboratif...ok. + ordonnancement préemptif...ok. + ordonnanceur round-robin...ok.

+ ordonnanceur à files de priorités...ok.

#### \$> Gestion de la mémoire

- + Implémentation de la pagination (MMU)
- + Implémentation des appels systèmes :
  - + sys\_mmap
  - + sys\_munmap
- + Implémentation partielle de l'isolation
  - + pile isolée uniquement
  - + pas d'isolation du kernel



# \$> Ordonnancement

- + Round-Robin simple
  - + famine impossible en préemptif
  - + garantie d'un quota de temps égal pour chaque process
- + Files de priorités
  - + peu gérer N niveaux de priorités
  - + famine impossible sauf aux limites
  - + pas de changement de priorité à l'exécution



## \$> Sortie écran --function

- + Utilisation du FrameBuffer
- + Géométrique de base
  - + traits
- + Image

- + PPM : image avec couleur
- + PGM : image noir et blanc
- + Texte
  - + police à chasse fixe (mono-space)
  - + police à chasse proportionnelle (non mono-space)
  - + multi-curseur





- + Fonctionnement des images
  - + PGM & PPM header:

FORMAT
LARGEUR HAUTEUR
NB\_COULEUR
Description des pixels...

P6 800 600 255 57j "QSd9è"éς'{ς\_jsKLDJÉ\$...

+ Intégration des images à l'édition des liens

\$>

#### \$> Sortie écran --font

- + Fonctionnement des polices
  - + Chargement d'une police à l'aide d'un script



#### > X BitMap

```
#define char_font_33_width 8
#define char_font_33_height 20
static char char_font_33_bits[] = {
    0x00, 0x00, 0x00, 0x00, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x00,
    0x03, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```



#### \$> Sortie écran --cursor

- + Fonctionnement des curseurs
  - + Valeurs courantes du X et du Y
- + Contraintes:
  - + Gestion du backspace (facile avec du mono-space)
  - + Gestion du retour à la ligne
- + Buffer des derniers caractères entrés
- + Zone inscriptible du curseur
  - + Retour à la ligne
  - + Effacement de la zone



#### \$> Entrée clavier

- Utilisation de la bibliothèque CSUD
- + Affichage des LEDs compatible 100%
- + Récupération des caractères compatibilité 50%
- + Fonctionnement pas optimal
  - + Pas d'interruption
  - + Temps de parcours très long
  - + Difficulté d'intégration



#### \$> Problèmes rencontrés

- + Globalement peu de problèmes de configuration
- + Trouver de la doc fiable n'est pas simple
- + Peu de partage de solution sur internet
  - + Pourtant résolution parfois simple
- + Impossible de debug facilement sur Raspberry
- + Configuration de QEMU :
  - + Impossible de gérer des périphériques (USB, clavier)
  - + Résolution de l'écran d'affichage



## \$> Évaluation du module

- + Ce qu'il manque dans le sujet ?
  - + quelques précisions concernant le fonctionnement des codes déjà fournis ou tout simplement des commentaires dans le code
- + Ce qu'il manque pendant les séances ? en dehors des séances ?
  - + en dehors des séances une sorte de wiki pour le projet qui pourrait être mis en place pour proposer des pistes et éventuellement fournir des exemples
- + Quelles fonctionnalités auriez voulu ?
  - + mutex et fork (manque de temps)
  - + jouer du son (manque de temps)
  - + shell et gestionnaire de fichier (manque de temps)
- Qu'est ce qu'il y a eu de bien ?

  Bien guidé avec le sujet pour commencer

#### \$> Démos

- + Affichage d'images
- + Saisie de texte au clavier
- + Démonstration des politiques d'ordonnancement
  - + Round-robin (simple)
  - + Files de priorités (à priorités)

## \$> Merci !



