



# Formation Comprendre la stack graphique sous Linux

Formation sur site, 2 jours

Dernière mise à jour: 12 May 2022

<b>Titre</b>	<b>Formation Comprendre la stack graphique sous Linux</b>
<b>Objectifs opérationnels</b>	<ul style="list-style-type: none"><li>• Être capable de comprendre les bases de l’affichage graphique: représentation des images et des couleurs, affichage de pixels, opérations sur les pixels.</li><li>• Être capable de comprendre le matériel lié à l’affichage graphique: composants du <i>pipeline</i> graphique, matériel pour le rendu et l’affichage graphique.</li><li>• Avoir une compréhension claire des composants de la pile logicielle pour le graphique dans le noyau Linux et de leurs rôles: TTY, sous-systèmes <i>frame-buffer</i> et DRM.</li><li>• Avoir une compréhension claire de la pile logicielle pour le graphique en espace utilisateur: DRM au niveau espace utilisateur, X.org, Wayland, OpenGL.</li></ul>
<b>Durée</b>	<b>Deux jours - 16 h</b>
<b>Méthodes pédagogiques</b>	<ul style="list-style-type: none"><li>• Présentations animées par le formateur: 75% de la durée de formation</li><li>• Démonstrations pratiques réalisées par le formateur: 25% de la durée de formation</li><li>• Version électronique des supports de présentation, des instructions et des données de travaux pratiques. Les supports sont librement disponibles sur <a href="https://bootlin.com/doc/training/graphics">bootlin.com/doc/training/graphics</a>.</li></ul>
<b>Formateur</b>	Un des ingénieurs mentionnés sur : <a href="https://bootlin.com/training/trainers/">https://bootlin.com/training/trainers/</a>
<b>Langue</b>	Présentations : Français Supports : Anglais
<b>Public visé</b>	Développeurs de systèmes multimédia utilisant le noyau Linux



<b>Pré-requis</b>	<ul style="list-style-type: none"><li>• <b>Expérience solide en programmation avec le langage C:</b> les participants doivent maîtriser l'utilisation de types de données et structures complexes, des pointeurs, pointeurs sur fonction et du pré-processeur C.</li><li>• <b>Expérience en développement bas-niveau sous Linux et avec les interfaces matérielles:</b> les participants doivent avoir une compréhension minimale de la gestion mémoire, de l'interaction avec des interfaces matérielles (registres, interruptions) et de l'interaction entre les applications user-space Linux et le noyau Linux (appels systèmes). Suivre la formation <i>Développement de pilotes de périphériques noyau Linux</i> de Bootlin, disponible sur <a href="http://bootlin.com/fr/formation/noyau-linux/">bootlin.com/fr/formation/noyau-linux/</a> permet de remplir ce pré-requis.</li><li>• <b>Niveau minimal requis en anglais: B1,</b> d'après le <i>Common European Framework of References for Languages</i>, pour nos sessions animées en anglais. Voir <a href="http://bootlin.com/pub/training/cefr-grid.pdf">bootlin.com/pub/training/cefr-grid.pdf</a> pour une auto-évaluation.</li></ul>
<b>Équipement nécessaire</b>	<p><b>Pour les sessions en présentiel dans les locaux de nos clients, notre client doit fournir:</b></p> <ul style="list-style-type: none"><li>• Projecteur vidéo</li><li>• Un grand moniteur</li><li>• Un tableau pour écrire</li></ul>
<b>Modalités d'évaluation</b>	Seuls les participants qui auront assisté à l'intégralité des journées de formation, et qui auront obtenu plus de 50% de réponses correctes à l'évaluation finale recevront une attestation individuelle de formation de la part de Bootlin.
<b>Handicap</b>	Les participants en situation de handicap qui ont des besoins spécifiques sont invités à nous contacter à l'adresse <a href="mailto:training@bootlin.com">training@bootlin.com</a> afin de discuter des adaptations nécessaires à la formation.



## Jour 1 - Matin

### Présentation - Représentation des images et des couleurs

- Lumière, pixels et images
- Échantillonnage, domaine de fréquence, aliasing
- Quantification et représentation des couleurs
- Espaces colorimétriques et canaux, canal alpha
- Sous-échantillonnage YUV et chroma
- Plans de données de pixels, ordre d'analyse
- Formats de pixels, codes FourCC codes, modificateurs

*Introduction aux notions de base utilisées pour représenter les images en couleur.*

### Présentation - Dessin des pixels

- Accès aux données de pixels et itération
- Concepts autour de la pixellisation
- Dessin de rectangles
- Dessin de gradients linéaires
- Dessin de disques
- Dessin de gradients circulaires
- Dessin de lignes
- Aliasing de lignes et de formes, dessin sub-pixel
- Cercles et coordonnées polaires
- Courbes paramétriques

*Comment accéder aux données de pixels en mémoire et dessiner des formes simples.*

### Présentation - Opérations sur les pixels

- Copie de région
- Alpha blending
- Keying de couleur
- Mise à l'échelle et interpolation
- Filtrage linéaire et convolution
- Filtres de floutage
- Dithering

*Notions de base autour du filtrage, avec des exemples d'utilisation très courants.*

### Démo - Dessin et opérations

- Exemples de dessin de divers types de formes et de régions
- Exemples d'opérations de base sur les pixels

*Illustration des concepts présentés au fur et à mesure.*



## Jour 1 - Après-midi

### Présentation - Vue d'ensemble des composants du pipeline et généralités

- Types d'implémentations de matériel graphique
- Mémoire graphique et buffers
- Pipelines graphiques
- Vue d'ensemble du matériel d'affichage, de rendu et de vidéo

*Présentation du matériel impliqué dans les pipelines graphiques.*

### Présentation - Matériel d'affichage

- Technologies d'affichage visuel : CRT, plasma, LCD, OLED, EPD
- Timings d'affichage, modes et EDID
- Interfaces d'affichage : VGA, DVI, HDMI, DP, LVDS, DSI, DP
- Bridges et transcodeurs

*Présentation du fonctionnement interne du matériel d'affichage.*

### Présentation - Spécificités du matériel de rendu

- Digital Signal Processors (DSPs)
- Accélérateurs matériels dédiés
- Graphics Processing Unit (GPUs)

*Description de l'architecture du matériel de traitement et de rendu.*

### Présentation - Intégration système, mémoire et performance

- Intégration graphique et mémoire
- Mémoire partagée pour les graphiques
- Contraintes et performance de la mémoire graphique
- Soulager le processeur en utilisant du matériel dédié au graphisme
- Conseils pour les performances graphiques

*Sujets autour de l'intégration système, la gestion de la mémoire et les performances.*



## Jour 2 - Matin

---

### Présentation - Pile d'affichage

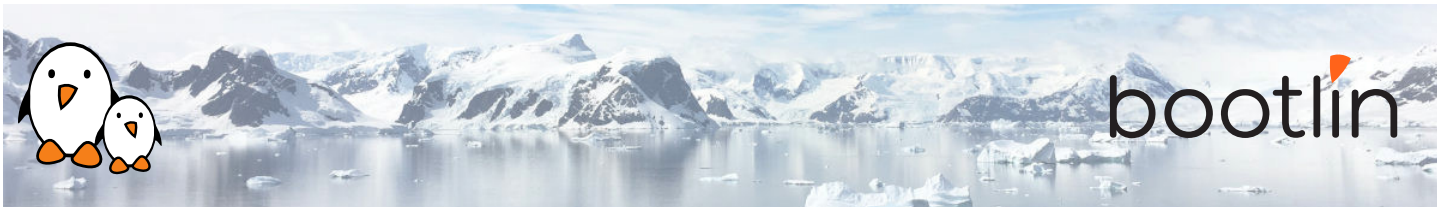
- Vue d'ensemble indépendante du système : noyau, affichage et rendu en espace utilisateur
- Vue d'ensemble de la partie dans le noyau Linux
- Vue d'ensemble de la partie bas niveau en espace utilisateur
- X Window et Wayland
- Bibliothèques graphiques haut niveau et environnements de bureau

*Présentation des composants nécessaires à un traitement graphique moderne, et comment ceux-ci sont répartis entre les espace noyau et utilisateur*

### Présentation - Aspects noyau TTY et device framebuffer

- Introduction au sous-système TTY de Linux
- Terminaux virtuels et graphiques
- Basculer entre terminaux virtuels et graphiques
- Vue d'ensemble de fbdev
- Opérations de base de fbdev
- Limitations de fbdev

*Comment les TTYs interagissent avec les graphiques sous Linux et brève présentation de fbdev et pourquoi ce composant n'est plus recommandé*



## Présentation - DRM dans le noyau

- Devices DRM
- Identification et fonctionnalités des pilotes DRM
- Maître DRM, "magic authentication"
- Gestion de la mémoire des DRM
- API "dumb buffer" de DRM KMS
- Modificateurs et FourCCs dans DRM
- Détection des ressources dans DRM KMS
- Modes DRM KMS
- Gestion de framebuffer dans DRM KMS
- Ancien système de configuration de DRM KMS et échange de pages
- Notification d'évènements dans DRM
- Propriétés d'objets dans DRM KMS
- DRM KMS atomic
- Rendu DRM
- Partage mémoire sans copie (dma-buf) avec DRM Prime
- Barrières d'objets DRM sync
- Débug et documentation dans DRM

*Une présentation complète de l'interface DRM.*

## Démo - Aspects noyau

- Terminaux virtuels et TTYs dans Linux
- Configuration des modes DRM KMS
- Visite guidée d'un pilote DRM KMS
- Visite guidée d'un pilote de rendu DRM

*Illustration du fonctionnement en espace noyau.*

## Jour 2 - Après-midi

### Présentation - Aspects X Window en espace utilisateur

- Protocole X11 et son architecture
- Extensions au protocole X11
- Architecture de Xorg et accélération
- Présentation des pilotes Xorg
- Accélération X11 et OpenGL : GLX et DRI2
- Utilisation de Xorg, intégration et configuration
- Principaux problèmes avec X11
- Débug et documentation de Xorg

*Présentation des tous les aspects de X11 et Xorg.*

### Présentation - Aspects Wayland en espace utilisateur

- Vue d'ensemble et paradigmes de Wayland
- Protocole Wayland et son architecture
- Détails sur le coeur du protocole Wayland
- Protocoles supplémentaires de Wayland
- Interface asynchrone de Wayland
- Intégration OpenGL de Wayland
- Statut et adoption de Wayland
- Débug et documentation de Wayland

*Une présentation approfondie de Wayland.*





## Présentation - Aspects Mesa 3D en espace utilisateur

- APIs de rendu 3D standardisées : OpenGL, OpenGL ES, EGL and Vulkan
- Vue d'ensemble de Mesa 3D
- Principaux détails d'implémentation de Mesa 3D
- Détails internes de Mesa 3D : Gallium 3D
- Détails internes de Mesa 3D : représentations intermédiaires
- Generic Buffer Management (GBM) dans Mesa 3D
- Point sur la prise en charge du matériel par Mesa 3D
- Mesa 3D comparée aux implémentations propriétaires
- Prise en charge du matériel par Mesa 3D : débog et documentation

*Présentation des APIs 3D et implémentation de Mesa 3D.*

## Démo - Aspects en espace utilisateur

- Visite guidée du code de Xorg
- Visite guidée du coeur du compositeur Wayland
- Exemples de clients Wayland
- Visite guidée du code de Mesa
- Exemples OpenGL et EGL

*Illustration des aspects en espace utilisateur et d'implémentations de clients et de serveurs.*