01B : Unix et appels système

420-3A6-ST : Programmation système

Dany Gagnon

Cégep de Saint-Jean-sur-Richelieu

Mercredi 26 août 2026

Table des matières

Deux repères dans les diapositives

Concept essentiel

L'étoile indique une diapositive ou un concept **particulièrement important** dans le cours.

Information supplémentaire

La loupe indique un complément utile pour approfondir la notion.

La question de la séance

Comment ces quelques lignes de C
finissent-elles par afficher du texte à l'écran ?

```c
#include <stdio.h>

int main(void) {
    puts("Bonjour, système!");
    return 0;
}
```

But de la séance

Comprendre comment un programme demande un service au noyau :

programme → appel système → changement de mode du CPU → noyau

Trois repères à conserver :

- distinguer Unix, Linux et Arch Linux ;
- reconnaître la frontière entre les modes utilisateur et noyau ;
- suivre une demande simple jusqu'à `write`.

Situer le cours

Ce que nous allons apprendre cette session

La programmation système étudie ce qu'un programme demande au système d'exploitation :

- créer et coordonner des processus ;
- manipuler des fichiers ;
- utiliser la mémoire ;
- communiquer et contrôler les accès.

Aujourd'hui : **observer une demande faite au système**.

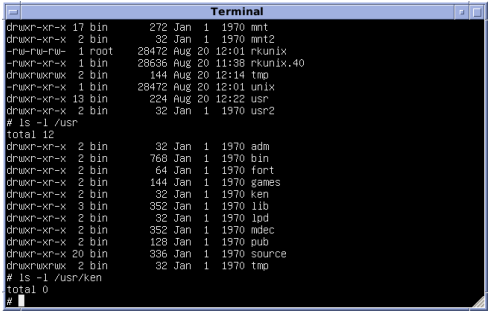
Le modèle Unix

Une idée née avec Unix

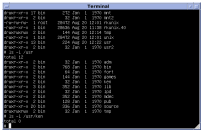
Unix propose une séparation simple :

- les programmes font leur travail en espace utilisateur ;
- le noyau contrôle les ressources et le matériel ;
- les programmes demandent des services au noyau.

Cette idée est encore au cœur de Linux.



Unix, Linux et Arch Linux



Unix

Une famille historique.

Linux

Un noyau de type Unix.

Arch Linux

Une distribution GNU/Linux.

À retenir

Arch Linux = **noyau Linux** + outils + applications

Linus Torvalds et le noyau Linux

Linus Torvalds

En 1991, Linus Torvalds commence à développer un nouveau **noyau**.

Ce projet devient Linux

Un noyau de type Unix, écrit comme un nouveau projet.

Torvalds n'a créé ni Unix ni Arch Linux. Son projet est le noyau Linux.

Linux est souvent là sans être visible

Web
92 % des sites ont un système de type Unix dont 62 % sous Linux
sites au système connu, août 2026

Téléphones Android
Un noyau basé sur Linux

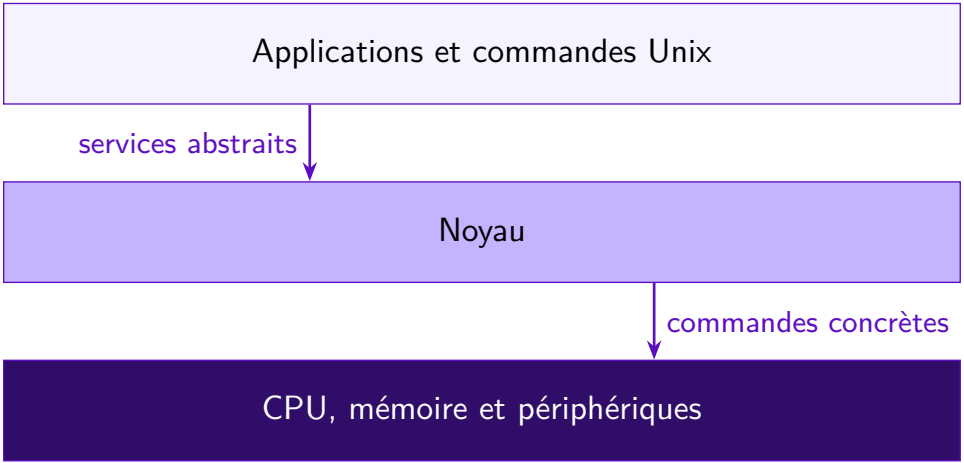
Noyau Linux

Superordinateurs
Linux sur les 500 systèmes du TOP500
juin 2026

Appareils embarqués
Routeurs, téléviseurs, systèmes automobiles et certains électroménagers

Même type de noyau, usages très différents.

Le noyau comme intermédiaire



Une application demande « écris ces octets ». Le noyau les achemine vers la ressource visée.

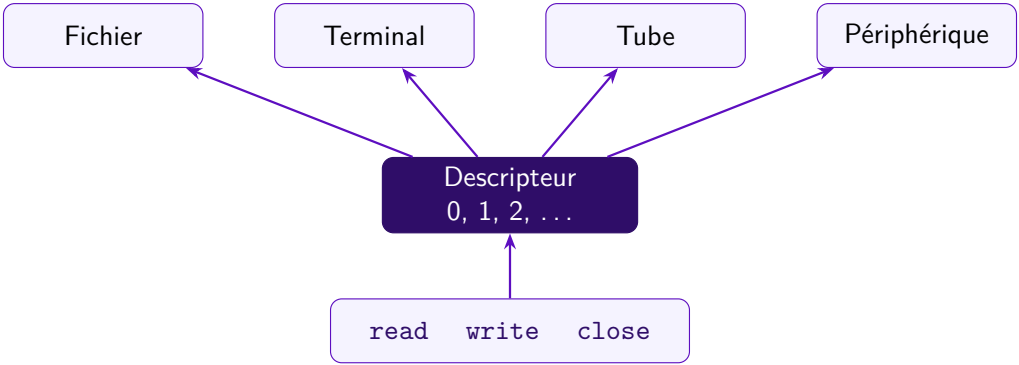
Une définition de travail

Système d'exploitation

Un logiciel qui fournit des services aux programmes et gère les ressources de la machine.

Ici, nous étudions surtout le **noyau**. Le shell, les commandes et l'interface graphique utilisent ses services.

Sous Unix et Linux, « tout est un fichier »



Plusieurs ressources se manipulent avec les mêmes appels système.

Le slogan n'est pas littéral. Il décrit une interface commune.

Le processeur ne connaît pas Unix

Le CPU sait :

- exécuter des instructions ;
- manipuler les registres et la mémoire ;
- réagir à certains événements matériels.

Le CPU ne sait pas ce qu'est :

- un utilisateur ;
- un fichier ;
- un processus ;

Le noyau construit ces concepts avec les mécanismes fournis par le matériel.

Le processeur ne connaît pas Unix

Le CPU sait :

- exécuter des instructions ;
- manipuler les registres et la mémoire ;
- réagir à certains événements matériels.

Le CPU ne sait pas ce qu'est :

- un utilisateur ;
- un fichier ;
- un processus ;

Le noyau construit ces concepts avec les mécanismes fournis par le matériel.

Question

Si une application et le noyau sont tous les deux du code, pourquoi l'application ne peut-elle pas tout faire ?

Modes d'exécution

Deux modes imposés par le processeur

Le processeur conserve un état qui indique son niveau de privilège.

Mode utilisateur

Les applications sont limitées. Certaines instructions et zones mémoire sont interdites.

Mode noyau

Le noyau peut configurer le matériel et utiliser les instructions privilégiées.

Si le mode n'est pas le bon, le CPU refuse physiquement l'opération.

Qui choisit le mode ?

1. Au démarrage, le processeur est en mode noyau.
2. Le noyau configure la machine.
3. Avant de lancer une application, il place le CPU en mode utilisateur.
4. Une transition contrôlée permet ensuite de revenir dans le noyau.

Une application ne peut pas simplement modifier le mode. Sinon, le privilège ne protégerait rien.

Le problème à résoudre

Un processus veut écrire dans le terminal.

- Il est en mode utilisateur.
- Écrire vers le terminal passe par un service du noyau.
- Un appel de fonction normal ne change pas le mode du CPU.

Il lui faut une entrée contrôlée : **l'appel système**.

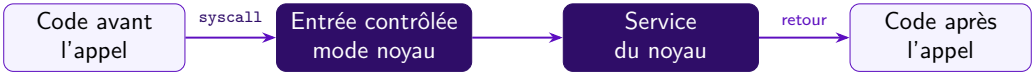
Ce que fait l'instruction spéciale

Lorsqu'un processus demande un appel système :

1. le CPU sauvegarde l'état minimal nécessaire à la transition ;
2. il passe en mode noyau et branche vers une entrée déterminée ;
3. le code d'entrée du noyau sauvegarde le reste du contexte requis ;
4. le noyau lit le numéro de l'appel et ses arguments.

Le matériel impose le point d'entrée. Le processus ne choisit pas une adresse arbitraire dans le noyau.

Une transition contrôlée



Au retour, le contexte est restauré et le CPU repasse en mode utilisateur.

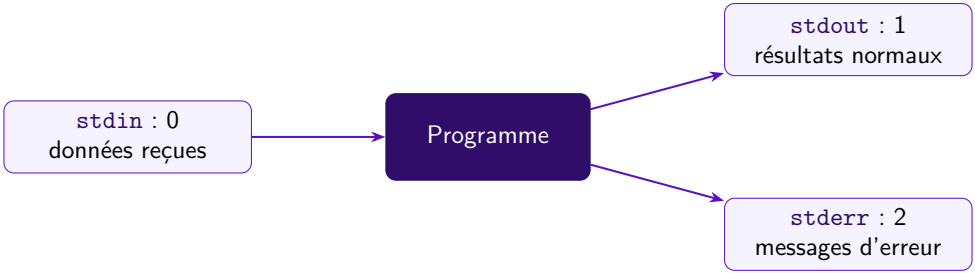
Appel de fonction ou appel système ?

	Appel de fonction	**Appel système**
Destination	adresse d'une fonction	entrée contrôlée du noyau
Privilège	mode inchangé	passage en mode noyau
Identification	adresse	numéro d'appel système
Coût	faible	plus élevé

Appels système Unix

Trois flux déjà ouverts

Un programme Unix commence normalement avec trois canaux :



Le shell relie habituellement ces flux au terminal. Il peut aussi les rediriger vers des fichiers.

Un appel système

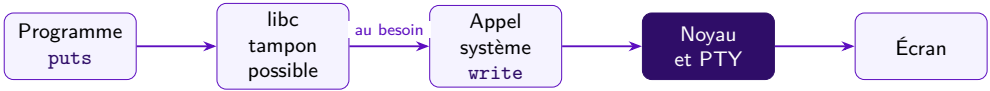
Un **appel système** est une demande contrôlée faite par un processus au noyau.

Dans notre exemple

Dans cette exécution, la bibliothèque C finit par demander au noyau d'écrire 19 octets sur `stdout`, le descripteur 1.

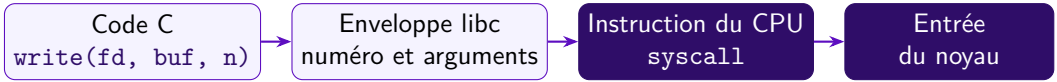
Le noyau vérifie la demande, effectue l'opération et retourne un résultat.

De puts à l'écran



`puts` peut garder les données en mémoire tampon avant de demander leur écriture.

L'enveloppe cache les détails du processeur



La libc prépare le numéro et les arguments, puis exécute l'instruction `syscall`.

Appeler l'enveloppe POSIX

```c
#include <unistd.h>

int main(void)
{
    const char message[] = "Bonjour, noyau!\n";
    ssize_t resultat = write(STDOUT_FILENO, message,
                             sizeof(message) - 1);
    return resultat == -1;
}
```

`write` est une fonction C déclarée dans `unistd.h`. La libc effectue la transition contrôlée.

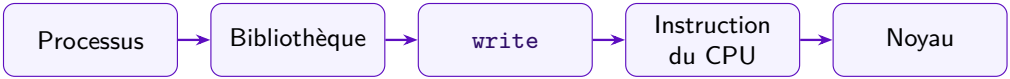
Commande, bibliothèque ou noyau ?

Les sections du manuel nous aident à situer un élément :

```
$ man 1 strace    # commande
$ man 3 puts      # fonction de bibliothèque
$ man 2 write     # appel système
```

Dans `man 2 write`, repérez le synopsis, la valeur de retour et la section *ERRORS*.

La chaîne de l'appel système



La fonction C cache les détails propres au processeur. `strace` nous laisse voir la demande et son résultat.

Préparer la machine virtuelle

Le résultat attendu

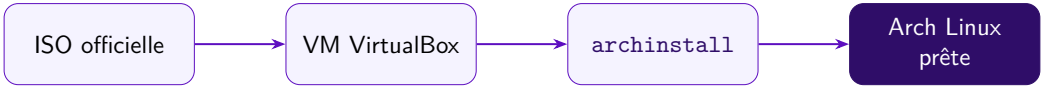
À la fin de l'installation, la VM doit :

- démarrer Arch Linux et ouvrir une session graphique ;
- accéder à Internet ;
- fournir GCC, Make, GDB, `strace` et le manuel ;
- servir de laboratoire commun pendant la session.

Un seul environnement de cours

Les démonstrations et les exercices seront préparés pour cette VM Arch Linux.

Le chemin jusqu'à une VM prête



Nous suivons ce chemin une seule fois, dans cet ordre.

1. Télécharger l'ISO officielle

1. Ouvrez `archlinux.org/download`.
2. Téléchargez l'ISO par un miroir canadien.
3. Conservez le fichier `archlinux-...-x86_64.iso`.

Une ISO est un disque d'installation

Ne la décompressez pas. VirtualBox la placera dans le lecteur optique virtuel.

2. Créer la VM dans VirtualBox

1. Cliquez sur **Nouvelle**.
2. Nommez la machine `3A6-Arch`.
3. Choisissez le type `Linux` et la version `Arch Linux (64-bit)`.
4. Sélectionnez l'ISO téléchargée.
5. Désactivez l'installation automatique si VirtualBox la propose.

3. Attribuer les ressources

Ressource	Choix du cours
Processeurs	2
Mémoire	4 Go
Disque virtuel	30 Go, allocation dynamique
Réseau	NAT
Démarrage	EFI activé

Restez dans la zone verte indiquée par VirtualBox.

4. Démarrer sur l'ISO

Au premier démarrage :

1. vérifiez que l'ISO Arch est dans le lecteur virtuel ;
2. lancez la VM ;
3. choisissez **Arch Linux install medium** ;
4. attendez l'invite `root@archiso`.

Le système exécuté à ce moment vient de l'ISO. Il n'est pas encore installé sur le disque virtuel.

5. Vérifier le réseau

Le réseau NAT de VirtualBox ressemble à une connexion filaire.

```
$ ping -c 3 archlinux.org
```

Trois réponses indiquent que l'installateur pourra télécharger les paquets.

Si le nom ne répond pas

Vérifiez que l'adaptateur réseau de la VM est activé et réglé à **NAT**.

6. Lancer l'installateur guidé

```
$ archinstall
```

Dans les menus :

- flèches pour déplacer la sélection ;
- Entrée pour ouvrir ou confirmer ;
- Espace pour cocher un élément ;
- Échap pour revenir au menu précédent.

7. Choisir la langue et les paramètres régionaux

Menu	**Choix recommandé**
Langue d'Archinstall	Français
Disposition du clavier	`cf`
Langue du système	`fr_CA.UTF-8`
Encodage	`UTF-8`

Testez les caractères `/`, `-`, `:` et `@` avant de continuer.

8. Choisir les miroirs

Menu	**Choix du cours**
Miroirs et dépôts	région **Canada**
Dépôts optionnels	aucun

Les miroirs fournissent les paquets téléchargés pendant l'installation et les mises à jour.

9. Préparer le disque virtuel

Dans **Configuration du disque** :

1. choisissez le partitionnement automatique ;
2. sélectionnez l'unique disque virtuel de 30 Go ;
3. choisissez **ext4** ;
4. laissez LVM et le chiffrement désactivés.

Le disque sélectionné sera effacé

Sa taille doit correspondre aux 30 Go créés dans VirtualBox.

10. Choisir le système de base

Menu	**Choix du cours**
Swap	activé, avec zram
Chargeur de démarrage	`systemd-boot`
Noyau	`linux`
Nom de machine	`3a6-arch`

Ces choix donnent une installation simple, sans composant spécialisé.

11. Créer le compte de travail

Dans **Authentification** :

1. nommez l'utilisateur `da` suivi de votre DA, par exemple `da1234567` ;
2. choisissez un mot de passe que vous retiendrez ;
3. accordez les droits **sudo** à cet utilisateur ;
4. laissez le compte root sans mot de passe direct.

`sudo` demandera le mot de passe de cet utilisateur.

12. Ajouter le bureau GNOME

Dans **Profil** :

1. choisissez **Desktop** ;
2. choisissez **GNOME** ;
3. choisissez le pilote **VirtualBox (open-source)** ;
4. conservez `gdm` comme écran de connexion.

GNOME fournit le bureau, les réglages et une application Terminal.

13. Régler l'audio et le réseau installé

Menu	**Choix du cours**
Applications → Audio	`PipeWire`
Configuration réseau	NetworkManager, backend par défaut
Bluetooth et impression	désactivés

NetworkManager préparera automatiquement la connexion virtuelle au prochain démarrage.

14. Ajouter les outils du cours

Conservez d'abord la configuration de `pacman` par défaut.

Dans **Paquets supplémentaires**, entrez :

```
base-devel gcc make gdb strace man-db man-pages git
```

- `gcc` compile les programmes C ;
- `make` automatise la compilation ;
- `strace` montre les appels système ;
- `man-db` et `man-pages` fournissent le manuel.

15. Régler l'heure du système

Menu	**Choix du cours**
Fuseau horaire	`America/Toronto`
Synchronisation NTP	activée

Le fuseau règle l'heure affichée. NTP garde l'horloge synchronisée par le réseau.

16. Relire avant d'installer

Le menu principal doit maintenant montrer :

- un disque configuré en ext4 ;
- le noyau `linux` et `systemd-boot` ;
- un utilisateur avec sudo ;
- le profil Desktop avec GNOME ;
- PipeWire et NetworkManager ;
- les paquets supplémentaires du cours.

Si une ligne obligatoire est vide, ouvrez-la et complétez-la avant de choisir **Installer**.

17. Lancer l'installation

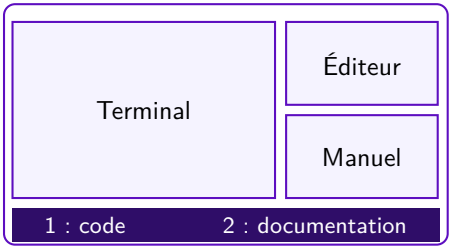
1. Sélectionnez **Installer** dans le menu principal.
2. Confirmez l'effacement du disque virtuel.
3. Laissez la VM ouverte pendant le téléchargement et l'installation.
4. À la fin, choisissez **Redémarrer le système**.

Pendant l'installation

N'éteignez pas la VM et ne fermez pas VirtualBox.

Pour aller plus loin : essayer i3

GNOME est un bureau complet. **i3** pave les fenêtres et favorise le clavier.



L'installer à côté de GNOME

```
sudo pacman -S --needed
xorg-server i3-wm i3status dmenu
```

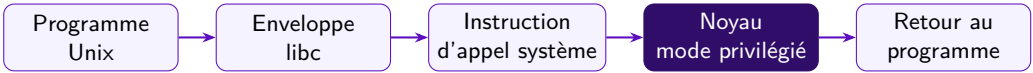
1. Déconnectez-vous de GNOME.
2. À l'écran de connexion, choisissez la session **i3**.
3. Au premier démarrage, choisissez `Super` comme touche principale.

`Super+Entrée` : terminal `Super+d` : applications `Super+Shift+e` : quitter

GNOME reste disponible depuis le même écran de connexion.

Synthèse

Retour à la question de départ



Dans notre exécution, `puts` mène à `write(1, ...)`, le noyau achemine les octets, puis le programme reprend.

Le matériel impose la frontière. Unix construit ses services au-dessus de cette frontière.