

Architecture des ordinateurs X

Frédéric Boulanger

CentraleSupélec



Système d'exploitation



Rôle

Le système d'exploitation fournit une abstraction du matériel :



Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs



Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques



Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques
- ▶ gestion des multi-cœurs



Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques
- ▶ gestion des multi-cœurs
- ▶ accès aux périphériques via les pilotes



Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques
- ▶ gestion des multi-cœurs
- ▶ accès aux périphériques via les pilotes

Il fournit également des services :

Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques
- ▶ gestion des multi-cœurs
- ▶ accès aux périphériques via les pilotes

Il fournit également des services :

- ▶ protocoles réseau

Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques
- ▶ gestion des multi-cœurs
- ▶ accès aux périphériques via les pilotes

Il fournit également des services :

- ▶ protocoles réseau
- ▶ communication et synchronisation entre processus

Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques
- ▶ gestion des multi-cœurs
- ▶ accès aux périphériques via les pilotes

Il fournit également des services :

- ▶ protocoles réseau
- ▶ communication et synchronisation entre processus
- ▶ protection entre processus

Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques
- ▶ gestion des multi-cœurs
- ▶ accès aux périphériques via les pilotes

Il fournit également des services :

- ▶ protocoles réseau
- ▶ communication et synchronisation entre processus
- ▶ protection entre processus
- ▶ protection entre utilisateurs

Rôle

Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques
- ▶ gestion des multi-cœurs
- ▶ accès aux périphériques via les pilotes

Il fournit également des services :

- ▶ protocoles réseau
- ▶ communication et synchronisation entre processus
- ▶ protection entre processus
- ▶ protection entre utilisateurs
- ▶ chargement des applications



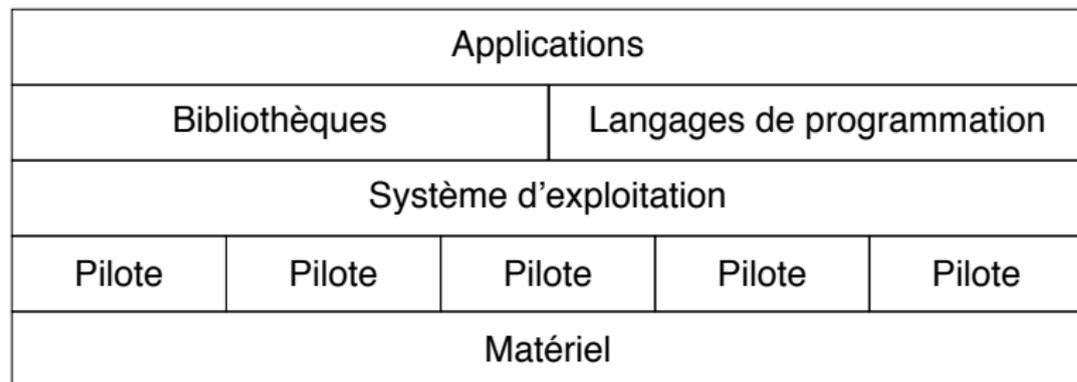
Le système d'exploitation fournit une abstraction du matériel :

- ▶ applications portables sur différents ordinateurs
- ▶ système de fichiers indépendant des périphériques
- ▶ gestion des multi-cœurs
- ▶ accès aux périphériques via les pilotes

Il fournit également des services :

- ▶ protocoles réseau
- ▶ communication et synchronisation entre processus
- ▶ protection entre processus
- ▶ protection entre utilisateurs
- ▶ chargement des applications
- ▶ langage de commandes

Structure



Sécurité

Le système d'exploitation s'appuie sur des mécanismes matériels pour assurer la sécurité :

- ▶ le processeur démarre en mode **privilégié**,
à une adresse où se trouve le code de chargement du système



Sécurité

Le système d'exploitation s'appuie sur des mécanismes matériels pour assurer la sécurité :

- ▶ le processeur démarre en mode **privilégié**, à une adresse où se trouve le code de chargement du système
- ▶ lorsque le système lance une application, il met le processeur en mode restreint



Sécurité

Le système d'exploitation s'appuie sur des mécanismes matériels pour assurer la sécurité :

- ▶ le processeur démarre en mode **privilégié**,
à une adresse où se trouve le code de chargement du système
- ▶ lorsque le système lance une application,
il met le processeur en mode restreint
- ▶ dans ce mode, certaines instructions sont interdites,
notamment celles qui contrôlent les adresses accessibles



Sécurité

Le système d'exploitation s'appuie sur des mécanismes matériels pour assurer la sécurité :

- ▶ le processeur démarre en mode **privilégié**,
à une adresse où se trouve le code de chargement du système
- ▶ lorsque le système lance une application,
il met le processeur en mode restreint
- ▶ dans ce mode, certaines instructions sont interdites,
notamment celles qui contrôlent les adresses accessibles
- ▶ il n'y a aucune instruction pour revenir en mode privilégié



Sécurité

Le système d'exploitation s'appuie sur des mécanismes matériels pour assurer la sécurité :

- ▶ le processeur démarre en mode **privilégié**, à une adresse où se trouve le code de chargement du système
- ▶ lorsque le système lance une application, il met le processeur en mode restreint
- ▶ dans ce mode, certaines instructions sont interdites, notamment celles qui contrôlent les adresses accessibles
- ▶ il n'y a aucune instruction pour revenir en mode privilégié
- ▶ le seul moyen d'y revenir est de restaurer le registre d'état en rendant la main au système : **appel système**

Sécurité

Le système d'exploitation s'appuie sur des mécanismes matériels pour assurer la sécurité :

- ▶ le processeur démarre en mode **privilégié**, à une adresse où se trouve le code de chargement du système
- ▶ lorsque le système lance une application, il met le processeur en mode restreint
- ▶ dans ce mode, certaines instructions sont interdites, notamment celles qui contrôlent les adresses accessibles
- ▶ il n'y a aucune instruction pour revenir en mode privilégié
- ▶ le seul moyen d'y revenir est de restaurer le registre d'état en rendant la main au système : **appel système**
- ▶ sauf bug dans le code du système, ce mécanisme garantit la protection des zones de mémoire et des périphériques contre les accès frauduleux.



Conclusion

En partant de transistors, nous avons construit :



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties

Nous avons vu :

Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties

Nous avons vu :

- ▶ comment utiliser les interruptions

Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties

Nous avons vu :

- ▶ comment utiliser les interruptions
- ▶ comment traduire les langages de programmation



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties

Nous avons vu :

- ▶ comment utiliser les interruptions
- ▶ comment traduire les langages de programmation
- ▶ comment fonctionne un pilote de périphérique

Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties

Nous avons vu :

- ▶ comment utiliser les interruptions
- ▶ comment traduire les langages de programmation
- ▶ comment fonctionne un pilote de périphérique

Sur ces bases, un système d'exploitation :



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties

Nous avons vu :

- ▶ comment utiliser les interruptions
- ▶ comment traduire les langages de programmation
- ▶ comment fonctionne un pilote de périphérique

Sur ces bases, un système d'exploitation :

- ▶ fournit une vision abstraite du matériel



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties

Nous avons vu :

- ▶ comment utiliser les interruptions
- ▶ comment traduire les langages de programmation
- ▶ comment fonctionne un pilote de périphérique

Sur ces bases, un système d'exploitation :

- ▶ fournit une vision abstraite du matériel
- ▶ permet d'écrire des applications pour des matériels différents



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties

Nous avons vu :

- ▶ comment utiliser les interruptions
- ▶ comment traduire les langages de programmation
- ▶ comment fonctionne un pilote de périphérique

Sur ces bases, un système d'exploitation :

- ▶ fournit une vision abstraite du matériel
- ▶ permet d'écrire des applications pour des matériels différents
- ▶ assure la protection des processus et des utilisateurs



Conclusion

En partant de transistors, nous avons construit :

- ▶ des portes logiques
- ▶ des fonctions arithmétiques
- ▶ des circuits séquentiels
- ▶ un microprocesseur (chemin de données + séquenceur)
- ▶ des périphériques d'entrées-sorties

Nous avons vu :

- ▶ comment utiliser les interruptions
- ▶ comment traduire les langages de programmation
- ▶ comment fonctionne un pilote de périphérique

Sur ces bases, un système d'exploitation :

- ▶ fournit une vision abstraite du matériel
- ▶ permet d'écrire des applications pour des matériels différents
- ▶ assure la protection des processus et des utilisateurs
- ▶ offre des services évolués grâce à des bibliothèques



FIN

