

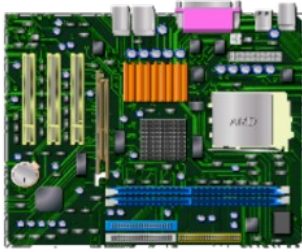


Sommaire

TP1 - Composants d'un ordinateurs.....	2
TP2 - Architecture de Von Neuman.....	3
TP3 - Système sur puce SoC.....	4
TP4 - Émulateur linux.....	5
TP5 - Arborescence.....	6
TP6 - Suppressions.....	7
TP7 - Éditeur de texte : nano.....	7
TP8 - Programme Python.....	7
TP9 - Gestion des droits.....	8
TP10 - Archives.....	9
TP11 - Identifiant PID des processus.....	9
TP12 - Ordonnancement avec une file.....	10
TP13 - Ordonnancement avec priorité.....	11
TP14 - Interblocage.....	12

TP1 - Composants d'un ordinateurs

- Identifier ci-dessous, les différents composants d'un ordinateur.



1



2



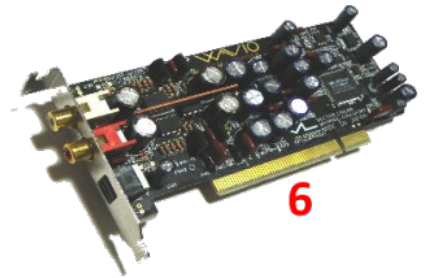
3



4



5



6



7



8

1.

2.

3.

4.

5.

6.

7.

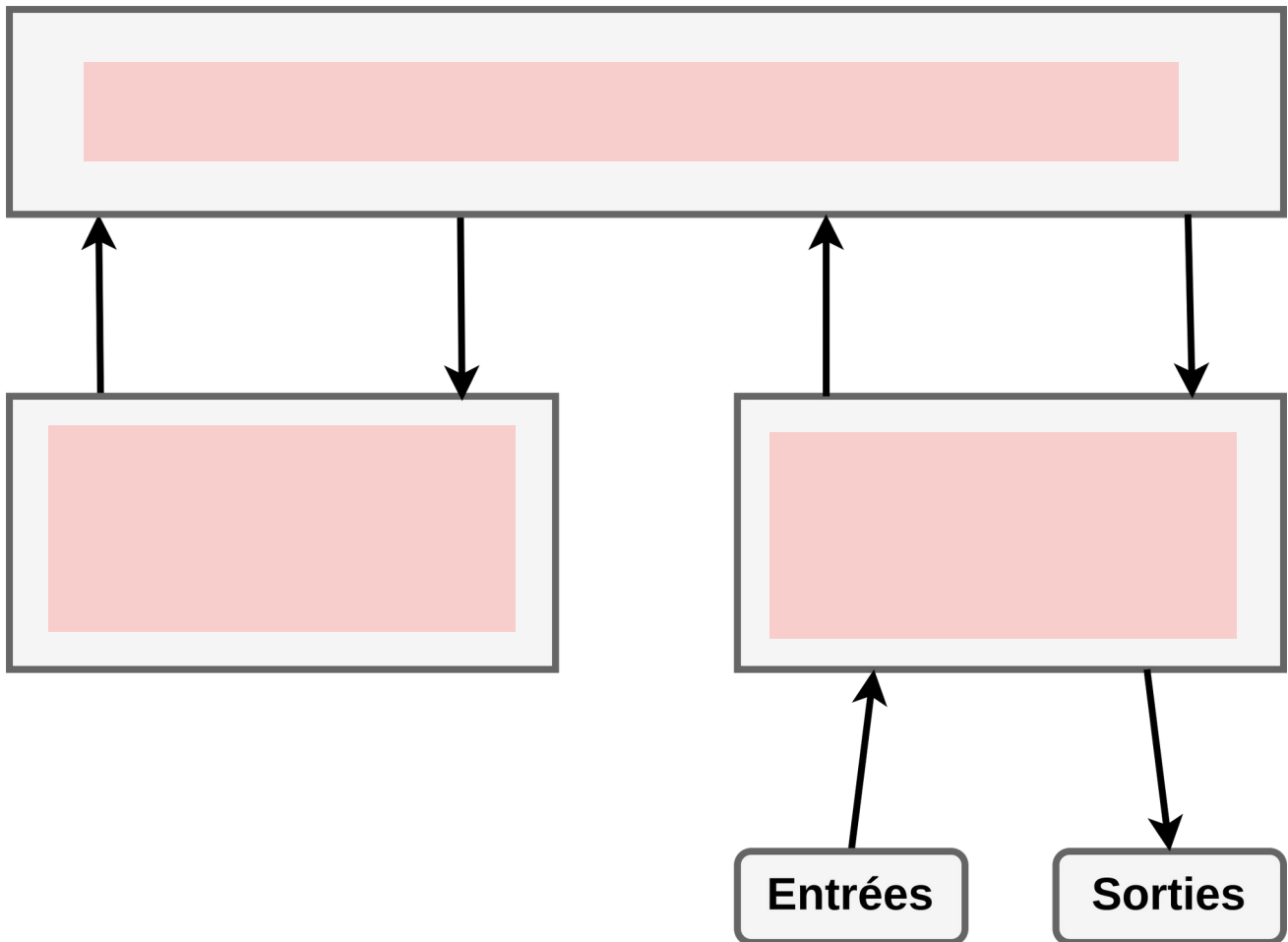
8.



Lever la main pour valider ce TP.

TP2 - Architecture de Von Neuman

- Compléter le schéma ci-dessous donnant l'architecture de Von Neuman, en nommant les différents éléments.



- Dans quel(s) élément(s) sont situés le « compteur de programme » (**CP** ou IP en anglais pour Instruction Pointer) et le « registre d'instruction » (**RI** ou IR en anglais pour Instruction Register). Préciser leurs rôles.

- CP est situé dans

Son rôle est

- RI est situé dans

Son rôle est



Lever la main pour valider ce TP.

TP3 - Système sur puce SoC

- **Définition** : « Un "système sur une puce", souvent désigné dans la littérature scientifique par le terme anglais "system on a chip" (d'où son abréviation SoC), est un système complet embarqué sur une seule puce ("circuit intégré"), pouvant comprendre de la mémoire, un ou plusieurs microprocesseurs, des périphériques d'interface, ou tout autre composant nécessaire à la réalisation de la fonction attendue. »

1. Citer un des avantages d'avoir plusieurs processeurs.

2. Citer un des avantages d'un circuit imprimé de petite taille.

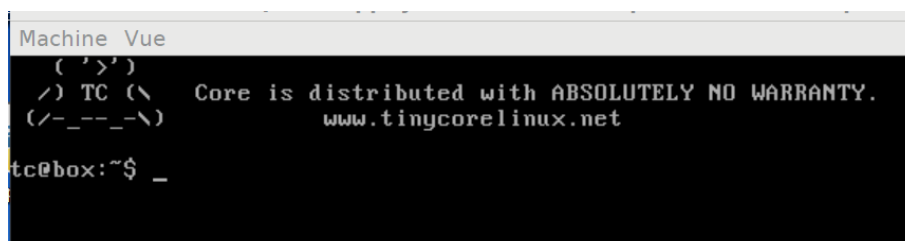
3. Citer un des inconvénients de cette miniaturisation.



Lever la main pour valider ce TP.

TP4 - Émulateur linux

- Sur votre compte, dans le dossier **NSI** , créer un sous-dossier **TP-Chapitre7** dans lequel on rangera les TP de ce chapitre.
- Récupérer le dossier **emul_linux.zip** en ressource, le copier dans votre dossier **TP-chapitre7** puis le décompresser ici.
- Lancer l'émulateur linux en cliquant sur le fichier **linux.vbs** . La fenêtre suivante s'affiche



```
Machine Vue
(  > )
/) TC (\\ Core is distributed with ABSOLUTELY NO WARRANTY.
(/-__-_-\\) www.tinycorelinux.net
tc@box:~$ _
```

On est maintenant connecté sous l'identifiant **tc** .

- Taper dans la console une commande pour déterminer quel est le répertoire dans lequel vous vous trouvez.

Commande :

Résultat :

- Taper dans la console une commande pour déterminer le contenu du répertoire dans lequel on se trouve.

Commande :

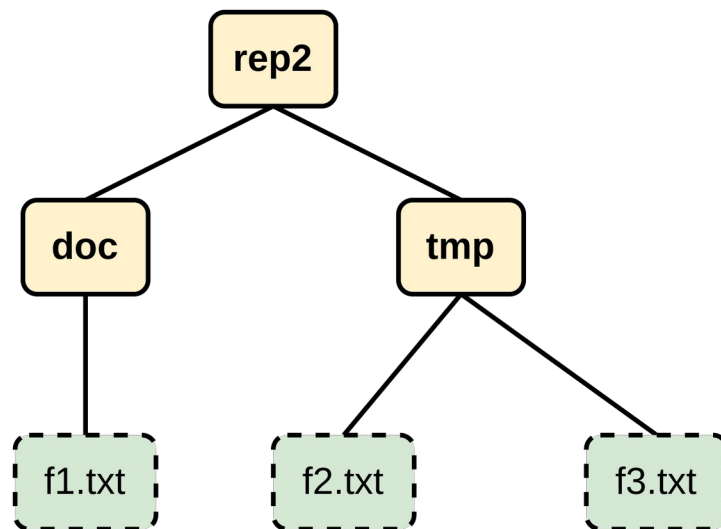
Nombre de sous-dossier :



Lever la main pour valider ce TP.

TP5 - Arborescence

- Dans le simulateur linux se déplacer dans le répertoire **rep1**.
- Explorer complètement le répertoire **rep1** à l'aide des commandes et .
- Sur une feuille de brouillon dessiner le contenu du répertoire **rep1** sous la forme d'une arborescence.
- A l'aide des commandes et créer un répertoire **rep2** dans le répertoire **tc** ainsi que le contenu de ce répertoire qui correspond à l'arborescence suivante :



- On considère qu'on est dans le répertoire courant **/home/tc**

Donner le chemin relatif du fichier **f1.txt** :

Donner le chemin absolu du fichier **f1.txt** :

 **Lever la main pour valider ce TP.**

TP6 - Suppressions

- Dans le simulateur linux supprimer le répertoire **rep3** ainsi que tout son contenu.
Pour cela il faut utiliser les commandes : et .



Lever la main pour valider ce TP.

TP7 - Éditeur de texte : nano

- Dans le simulateur linux se déplacer dans le répertoire **rep4**.
- A l'aide de l'éditeur **nano** ouvrir le fichier **identite.txt** et le compléter.
- Dans le répertoire **rep4** créer un sous-répertoire nommé **ID** et déplacer le fichier **identite.txt** à l'intérieur.
- Dans le répertoire **rep4** créer un autre sous-répertoire nommé **DOC** puis dans ce répertoire créer un fichier **bonjour.txt** dans lequel on va écrire 10 fois la phrase :
"Bonjour tout le monde !"
Pour cela utiliser : copier (Alt 6) et coller (Ctrl u).



Lever la main pour valider ce TP.

TP8 - Programme Python

- Dans le simulateur linux se déplacer dans le répertoire **rep5**.
- A l'aide de l'éditeur **nano** corriger le programme Python dans le fichier **programme.py**.
- Exécuter ce programme en tapant la ligne de commande suivante :
python programme.py



Lever la main pour valider ce TP.

TP9 - Gestion des droits

- Dans le simulateur linux se déplacer dans le répertoire **rep6** .
- Déterminer la notation octale des droits du fichier **file8.txt** : .
- A l'aide de l'éditeur **nano** compléter le fichier **file8.txt** avec votre nom et votre prénom.
Quel message d'erreur apparaît lorsque vous enregistrer le fichier ? :

- Modifier les droits pour qu'ils correspondent à la capture ci-dessous :

```
-rw-r--r--  1 tc      staff    15 Apr 30 16:17 file8.txt
```

- Quelle est la notation octale correspondante ? : .
- Maintenant compléter le fichier avec votre nom et votre prénom.



Lever la main pour valider ce TP.

TP10 - Archives

- Dans le simulateur linux en utilisant la commande extraire l'archive nommée **rep7.tar** .
- Explorer le répertoire **rep7** ainsi créé et dessiner sur une feuille de brouillon l'arborescence donnant son contenu .
- Créer une archive compressée nommée **rep7.tar.gz** .



Lever la main pour valider ce TP.

TP11 - Identifiant PID des processus

- Dans le simulateur linux taper la commande **ps -o pid,ppid,stat,cmd** .
- L'identifiant du processus correspondant à la commande **ps** est : .
- L'identifiant du processus parents du processus de la commande **ps** est : .
- Identifier la commande correspondant au processus parents du processus de la commande **ps** : .



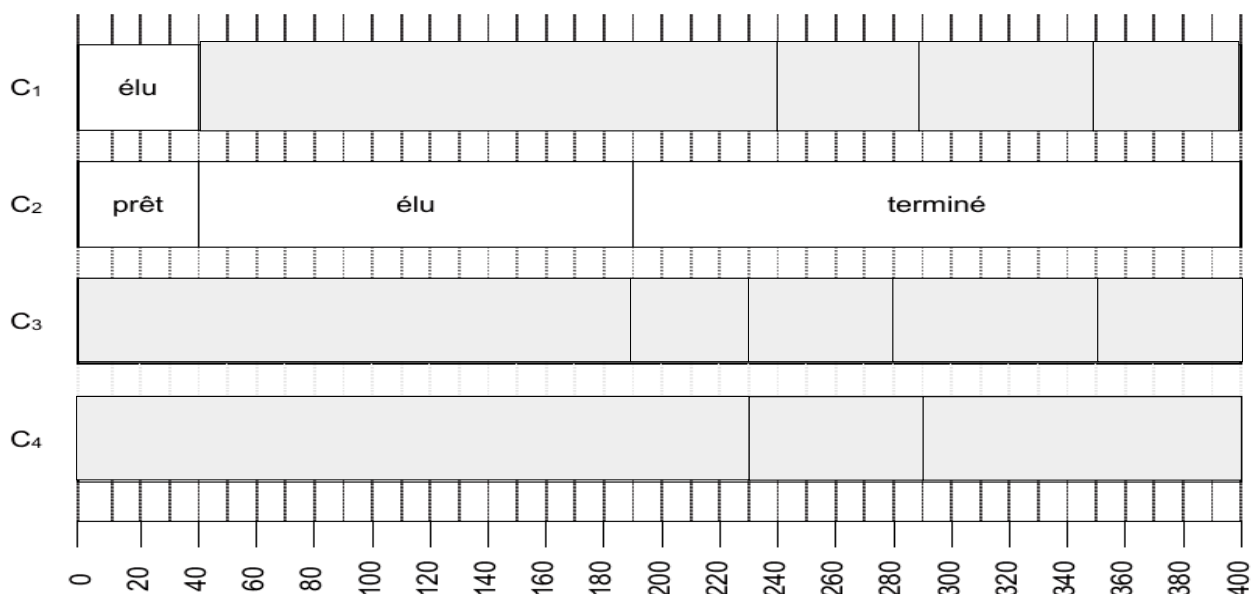
Lever la main pour valider ce TP.

TP12 - Ordonnancement avec une file

On suppose que quatre processus **C1**, **C2**, **C3** et **C4** sont créés sur un ordinateur, et qu'aucun autre processus n'est lancé sur celui-ci, ni préalablement ni pendant l'exécution des quatre processus. L'ordonnanceur, pour exécuter les différents processus prêts, les place dans une structure de données de type **file**. Un processus prêt est enfilé et un processus élu est défilé.

- Parmi les propositions suivantes, cocher celle qui décrit le fonctionnement des entrées/sorties dans une file :
 - ☐ Premier entré, dernier sorti
 - ☐ Premier entré, premier sorti
 - ☐ Dernier entré, premier sorti
- On suppose que les quatre processus arrivent dans la file et y sont placés dans l'ordre **C1**, **C2**, **C3** et **C4**.
 - Les temps d'exécution totaux de **C1**, **C2**, **C3** et **C4** sont respectivement 100 ms, 150 ms, 90 ms et 60 ms.
 - Après 40 ms d'exécution, le processus **C1** demande une opération d'écriture disque, opération qui dure 200 ms. Pendant cette opération d'écriture, le processus **C1** passe à l'état bloqué.
 - Après 40 ms d'exécution, le processus **C3** demande une opération d'écriture disque, opération qui dure 50 ms. Pendant cette opération d'écriture, le processus **C3** passe à l'état bloqué.

Sur la frise chronologique ci-dessous, les états du processus **C2** sont donnés. Compléter la frise avec les états des processus **C1**, **C3** et **C4**.



Lever la main pour valider ce TP.

TP13 - Ordonnancement avec priorité

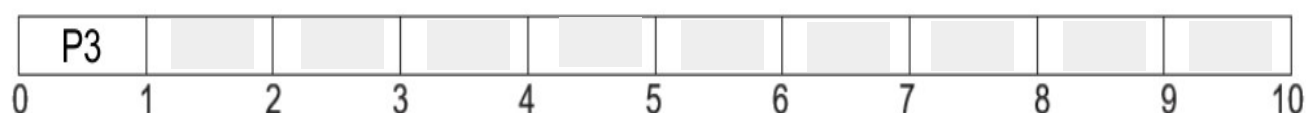
Un processeur choisit à chaque cycle d'exécution le processus qui doit être exécuté. Le tableau ci-dessous donne pour trois processus P1, P2, P3 :

- la durée d'exécution (en nombre de cycles),
- l'instant d'arrivée sur le processeur (exprimé en nombre de cycles à partir de 0),
- le numéro de priorité.

Le numéro de priorité est d'autant plus petit que la priorité est grande. On suppose qu'à chaque instant, c'est le processus qui a le plus petit numéro de priorité qui est exécuté, ce qui peut provoquer la suspension d'un autre processus, lequel reprendra lorsqu'il sera le plus prioritaire.

Processus	Durée d'exécution	Instant d'arrivée	Numéro de priorité
P1	3	3	1
P2	3	2	2
P3	4	0	3

Compléter le tableau ci-dessous et indiquer dans chacune des cases le processus exécuté à chaque cycle.



Lever la main pour valider ce TP.

TP14 - Interblocage

1. Un processeur choisit à chaque cycle d'exécution le processus qui doit être exécuté. On considère trois processus P1, P2, P3 que le processeur doit exécuter. Parmi les scénarios suivants, lequel provoque un interblocage ? Justifier en dessinant un schéma sur une feuille de brouillon.

Scénario 1	Scénario 2	Scénario 3
P1 acquiert R1	P1 acquiert R1	P1 acquiert R1
P2 acquiert R2	P2 acquiert R3	P2 acquiert R2
P3 attend R1	P3 acquiert R2	P3 attend R2
P2 libère R2	P1 attend R2	P1 attend R2
P2 attend R1	P2 libère R3	P2 libère R2
P1 libère R1	P3 attend R1	P3 acquiert R2

Le scénario qui provoque un interblocage est le scénario .

2. On trouvera ci-dessous deux programmes rédigés en pseudo-code. Verrouiller un fichier signifie que le programme demande un accès exclusif au fichier et l'obtient si le fichier est disponible.

Programme 1	Programme 2
Verrouiller fichier_1 Calculs sur fichier_1 Verrouiller fichier_2 Calculs sur fichier_1 Calculs sur fichier_2 Calculs sur fichier_1 Déverrouiller fichier_2 Déverrouiller fichier_1	Verrouiller fichier_2 Verrouiller fichier_1 Calculs sur fichier_1 Calculs sur fichier_2 Déverrouiller fichier_1 Déverrouiller fichier_2

- a. En supposant que les processus correspondant à ces programmes s'exécutent simultanément (exécution concurrente), expliquer le problème qui peut être rencontré en dessinant un schéma sur une feuille de brouillon.
- b. Proposer une modification du programme 2 permettant d'éviter ce problème.



Lever la main pour valider ce TP.