

## Questions de cours [6 pts]

1. Linux dispose naturellement de trois stratégies d'ordonnancement des tâches (SCHED\_OTHER, SCHED\_FIFO et SCHED\_RR). Quel est la stratégie par défaut de linux ? Quelle différence faites vous entre les stratégies SCHED\_OTHER et SCHED\_FIFO/RR ? [2 pts]

SCHED\_OTHER est la stratégie d'ordonnancement des tâches par défaut de linux.

SCHED\_OTHER est une stratégie d'ordonnancement équitable des tâches (temps partagé).

SCHED\_FIFO/RR permet d'ordonner l'ordonnancement des tâches en fonction de leur priorité.

Les tâches utilisant l'ordonnanceur SCHED\_FIFO/RR sont prioritaires devant celles utilisant SCHED\_OTHER.

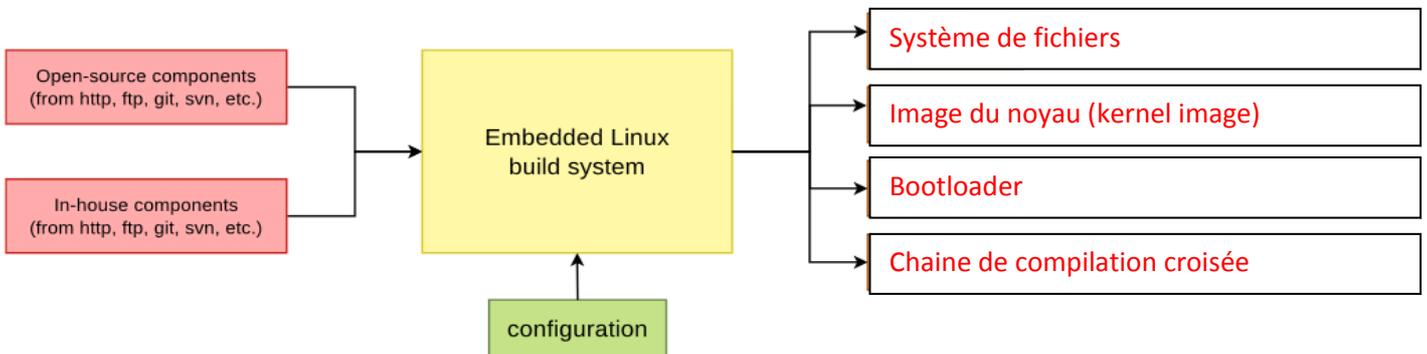
2. Que permet d'obtenir l'installation de l'extension XENOMAI sur un système linux ? [1 pts]

L'installation de l'extention XENOMAI sur un système lui confère des propriétés temps réel dur non certifiable. Linux devient une simple tâche de fond de Xenomai et peut être préempté.

3. Qu'est ce qu'un build system ? Donnez le nom du build system que vous avez utilisé en TP. [1 pts]

Un build system est un ensemble de logiciel permettant de construire un système d'exploitation et des outils de développement sur mesure pour une cible donnée. En TP, on a utilisé Buildroot.

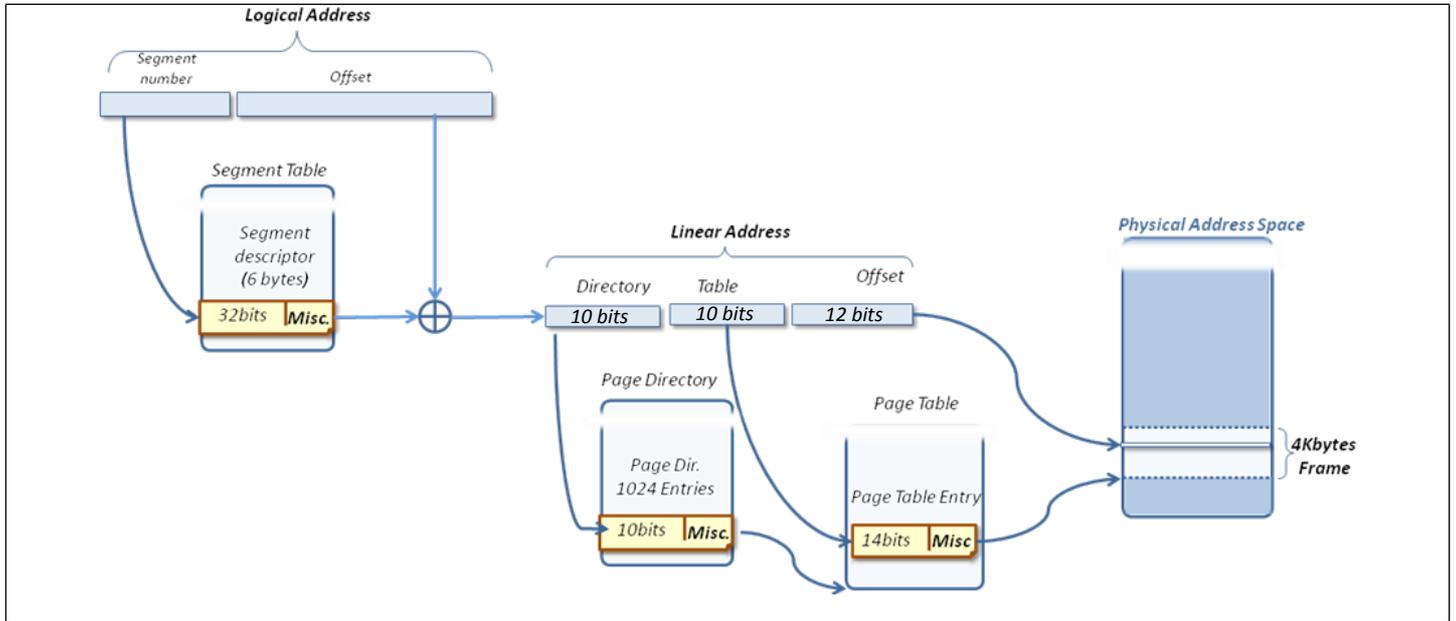
4. Quels sont les éléments construits par le build system. Complétez le schéma ci-dessous : [2 pts]



# Gestion de mémoire dans un système d'exploitation [6 pts]

Soit un système de gestion de mémoire gérée de manière segmentée et paginée avec double niveau de pagination.

- La taille de la mémoire physique est de 64 Mo (1mot = 1 octet).
- Un processus peut avoir au plus 256 segments.
- Chaque segment peut adresser au plus 16 Mo.
- La taille d'une page est fixée à 4 ko.



1. Quel est le format des adresses logiques ? Expliquez. [1 pts]

Une adresse logique est composée d'un n° de segment et d'un décalage.

- Le numéro de segment doit pouvoir représenter 256 segments => 8 bits
- Le décalage doit pouvoir adresser un espace de 16 Mo => 24 bits

2. Quel est le format d'une adresse physique ? Expliquez. [1 pts]

L'espace d'adressage physique est de 64 Mo => 26 bits.

3. Quelle est la taille de l'espace d'adressage virtuelle ? [1 pts]

1 adresse virtuelle = 32 bits donc espace d'adressage de 4 Go ou 256 segments de 16 Mo au max = 4 Go.

Soit un processus muni de la table des segments suivante :

Segment	Base	Limite
00	00 BE 0A 00	10 00
01	00 BE 23 D1	02 FF
02	00 BE 00 DA	03 61
03	00 BE 1A 26	05 07
04	00 BE 0F F0	10 00

du répertoire de pages suivant :

Répertoire	Table	Active
0	0	1
1	3	0
2	1	1
3	2	0

Et de deux tables de pages :

N°	@Page	@Cadre	Active	Libre
0	2A0	23 40	1	0
1	2A1	05 BB	1	0
2	2A2	00 00	0	0
3	2A3	14 E0	1	0
4				1
				1
1023				1

Table 0

N°	@Page	@Cadre	Active	Libre
0	3E0	27 FD	1	0
1	3E1	12 A0	0	0
2	3E2	00 00	0	0
3	3E3	3A F6	1	0
4				1
				1
1023				1

Table 1

4. Quelle est l'adresse linéaire correspondante à l'adresse logique **0x020000F0** [1 pts]

- N° de segment : 2 => adresse de base : **0x00BE00DA**
- Décalage : **0x0000F0** bien inférieur à la limite du segment (**0x0361**) pas d'erreur de dépassement
- Adresse linéaire = **0x00BE00DA + 0x0000F0 = 0x00BE01CA**

5. Quelle est l'adresse physique correspondante à l'adresse logique **0x020000F0** [1 pts]

Adresse logique **0x020000F0** <=> Adresse linéaire **0x00BE01CA**

0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0
10 bit = Répertoire : 2										10 bits = Table : 3E0										12 bits = Décalage : 1CA										

- L'entrée 2 du répertoire de page indique la table de pages 1.
  - L'entrée **0x3E0** de cette table indique le cadre **0x27FD**, qui correspond aux 14 bits de poids fort de l'adresse physique
  - Le décalage est égal à **0x1CA**, qui correspond aux 12 bits de poids faible de l'adresse physique
  - L'adresse physique correspondante : **0x27FD1CA**
6. Quelle quantité de mémoire physique occupe le processus. [1 pts]
- 1 répertoire de table contenant 4 numéros de table (de 0 à 3) et leur état d'activité (0 ou 1).
  - Numéro de table codé sur 10 bits et état d'activité sur 1 bit. Ces informations nécessitent 2 octets.
  - Quantité de mémoire physique occupée par le répertoire de table :  $4 \times 2 = 8$  octets (**négligeable**)
  - 2 tables contenant 1024 entrées chacune.
  - Numéro de cadre codé sur 14 bits + 1 bit d'activité + 1 bit de liberté : Nécessite 2 octets
  - Quantité de mémoire physique occupée par les tables :  $2 \times 2 \times 1024 = 4$  ko
  - 5 pages actives et une page occupe 4 ko :  $5 \times 4 = 20$  ko
  - Occupation totale en mémoire physique : **24 ko**

Rappel : puissances de 2

N	2 <sup>N</sup>	N	2 <sup>N</sup>	N	2 <sup>N</sup>	N	2 <sup>N</sup>
0	1	8	256	16	64k	24	16M
1	2	9	512	17	128k	25	32M
2	4	10	1024 = 1k	18	256k	26	64M
3	8	11	2k	19	512k	27	128M
4	16	12	4k	20	1M	28	256M
5	32	13	8k	21	2M	29	512M
6	64	14	16k	22	4M	30	1G
7	128	15	32k	23	8M	31	2G