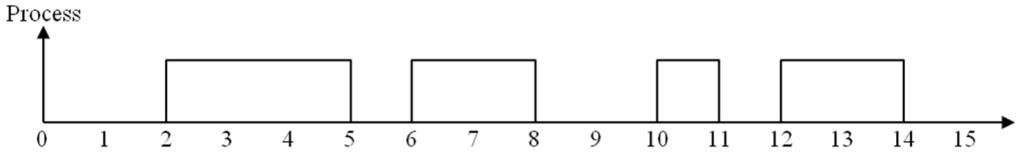
ORDONNANCEMENT DANS LES SYSTEMES CLASSIQUES

Il est recommandé d'utiliser un tableur pour réaliser les différents calculs. On rappelle que les supports de cours sont disponibles à http://mathieu.delalandre.free.fr/teachings/realtime.html

Paramètre des processus et exécution

Soit le diagramme d'occupation du processeur suivant, décrivant l'exécution du processus P. Les créneaux hauts correspondent aux cycles actifs (i.e. ou P est exécuté par le CPU).



Considérant une date de réveil $r_0 = 1$, indiquez les paramètres : capacité du processus, date de démarrage, date de fin, temps d'attente et temps de réponse. Donnez également sur l'intervalle de temps considéré (i.e. pour tout t) les valeurs de la capacité résiduelle C(t), du temps d'exécution T(t), du temps CPU ouvert ou âge du processus E(t) et du temps d'attente WT(t). Indiquez finalement les différents « CPU Burst » sur ce chronogramme d'exécution en précisant leur nombre, positions, et durées. On considéra ici tout changement de contexte comme dû à des interruptions I/O.

2. Algorithmes à base de priorités et optimaux

2.1. Algorithmes à priorités statiques / non-préemptifs

Soit le tableau de tâches suivant:

	r_0	C	P
A	0	60	3
B	0	100	2
C	0	40	4
D	0	20	1
E	0	80	5

Donnez les diagrammes d'occupation du processeur pour les algorithmes d'ordonnancement FCFS, SJF non-préemptif (i.e. SPN) et « Priority Scheduling » non-préemptif. Dans le cas du FCFS, on considéra les id processus (A à E) comme ordre des processus dans la queue d'exploitation. Indiquez les temps d'attente de chacune des tâches et le temps d'attente moyen pour chacun des algorithmes.

2.2. Algorithmes à priorités statiques / préemptifs

Soit le tableau de tâches suivant:

	r_0	C	P
A	0	4	4
B	3	6	3
C	4	3	2
D	8	5	5
E	9	2	1

Donnez le diagramme d'occupation du processeur pour l'algorithme d'ordonnancement « Priority Scheduling » préemptif sur l'intervalle t = [0-20]. Indiquez les temps d'attente de chacune des tâches et le temps d'attente moyen.

2.3. Algorithmes à priorités dynamiques / non-préemptifs

Soit le tableau de tâches suivant:

	r_0	C
A	0	4
B	2	2
C	1	6
D	6	2
E	3	3

Donner le diagramme d'occupation du processeur pour l'algorithme d'ordonnancement HRRN sur l'intervalle t = [0-17]. Indiquez les temps d'attente de chacune des tâches et le temps d'attente moyen sur l'intervalle considéré.

2.4. Algorithmes à priorités dynamiques / préemptifs

A partir du tableau de tâche donné précédemment en 2.2., donnez le diagramme d'occupation du processeur pour l'algorithme d'ordonnancement SJF préemptif (i.e SRT) sur l'intervalle t = [0-20]. En cas d'égalité du critère d'ordonnancement, vous départagerez en appliquant la règle suivante: le processus en cours garde la main. Indiquez les temps d'attente de chacune des tâches et le temps d'attente moyen.

Soit le tableau de tâches suivant:

	r_0	C	P
A	0	∞	3
B	3	∞	5
C	4	∞	4
D	8	∞	2

Donner le diagramme d'occupation du processeur pour l'algorithme d'ordonnancement « Dynamic Priority Scheduling » sur l'intervalle t = [0-20].

2.5. Prédiction temporelle

On considère deux processus A, B au sein d'un système interactif, ou l'ordonnancement des tâches est géré par technique de prédiction temporelle via l'algorithme SRT (Shortest Remaining Time). Soit le tableau suivant des durées d'exécution CPU (i.e. burst time) relevées entre chaque changement de contexte pour A, B.

t_i	t_0	t_1	t_2	t_3
A	15	21	26	24
B	21	15	13	26

Définissez les résultats de prédiction temporelle pour ces deux processus, sur les « burst » 0 à 3, considérant les paramètres suivants: $T_0 = 17$ pour A, $T_0 = 18$ pour B et $\alpha = 0.3$ pour A, B. Considérant ensuite le tableau d'événements suivant, ou chaque événement correspond à une transition de processus de l'état « blocked » vers « ready / running » (i.e. les processus concernés quittent la queue de blocage vers la queue d'exploitation), définissez la séquence d'exécution en précisant les informations suivantes: états des queues de blocage et d'exploitation, processus en cours d'exécution, valeurs de prédiction des processus en état « ready/running ». Pour l'établissement de cette séquence vous appliquerez la règle suivante : à chaque événement, les processus sont concurrents pour l'accès CPU (e.g. si B relâché, A est en cours d'exécution et vice-et-versa).

	1	**2**	**3**	**4**	**5**	**6**	**7**
Evénements	B	A	B	A	A, B	A	B