

Licence d'Informatique L3
Introduction aux Systèmes et Réseaux

Plan des TD et TP

TD1	<p>L'objectif de ce TD est d'approfondir les notions relatives aux processus et de les appliquer au cadre spécifique d'Unix.</p> <p>Points à développer :</p> <ul style="list-style-type: none">– opérations de création (<code>fork()</code>) et de fin (<code>exit()</code>), généalogie des processus.– relations père-fils avec <code>wait()</code>.– utilisation de <code>execve()</code> ; structure de la mémoire d'un processus Unix. <p>Ces notions seront appliquées dans le TP n°1.</p>	TP1 (1h30)	<p>Utilisation pratique de <code>fork()</code>, <code>exit()</code>. Structures généalogiques en arbre, râteau, etc. Utilisation de <code>wait()</code>.</p> <p>Lancement d'un processus avec un programme, passage de paramètres, variables d'environnement : utilisation de <code>execve()</code>.</p>
TD2	<p>L'objectif de ce TD est d'introduire une vue élémentaire des aspects de "bas niveau" des processus, pour faire notamment le lien avec l'enseignement d'ALM.</p> <p>Points à développer :</p> <ul style="list-style-type: none">– mise en œuvre des processus sur un processeur, ordonnancement.– relations entre interruptions (vues en ALM) et processus.– relations entre processus au moyen de signaux.– utilisation élémentaire des signaux. <p>Les notions sur les signaux seront appliquées dans le TP n°2.</p>	TP2 (1h30)	<p>Manipulation des signaux en Unix. Applications simples (réveil, mise au point, traitement des interruptions du terminal).</p> <p>Mesure du temps (<code>times()</code>).</p>
TD3	<p>L'objectif de ce TD est de reprendre et de compléter les notions sur la vie d'un programme, avec notamment les notions de bibliothèques et de liaison dynamique. On examinera également le fonctionnement d'un interprète, avec application au <i>shell</i> Unix. Ces notions seront appliquées dans le TP n°3.</p>	TP3 (1h30)	<p>Réalisation d'un mini-<i>shell</i>. Ce travail sera prolongé par une APNEE (projet non encadré, évalué).</p>

TD4	<p>L'objectif de ce TD est de poursuivre le thème de la communication entre processus, déjà entamé à propos des signaux, en introduisant la communication via les tubes (<i>pipes</i>) et les fichiers. Au passage, on montrera l'analogie entre tubes et fichiers et on introduira des exemples élémentaires de communication client-serveur. Points à développer :</p> <ul style="list-style-type: none"> - utilisation des descripteurs de fichiers, - communication par flots de données : tubes, FIFOs. <p>Ces notions seront appliquées dans le TP n°4.</p>	TP4 (1h30)	<p>Communication entre processus par tubes, au niveau du langage de commande, puis au niveau des appels systèmes. Tubes nommés (<i>fifo</i>s). Accès aux fichiers : <code>open()</code>, <code>read()</code>, <code>write()</code>, <code>close()</code>; utilisation dans des cas simples de <code>fcntl()</code> pour le verrouillage (mode exclusif).</p>
TD5	<p>L'objectif de ce TD est d'examiner deux aspects de la gestion de la mémoire. 1) Utilisation du couplage de fichiers en mémoire virtuelle (<code>mmap()</code>). 2) Allocation de mémoire : réalisation des opérations <code>malloc()</code> et <code>free()</code> dans une mémoire virtuelle simulée, avec la technique <i>first fit</i>.</p>		
		TP5 (1h30)	<p>Introduction aux <i>sockets</i> Unix en mode connecté (TCP). Manipulation des primitives (à travers une bibliothèque fournie). Utilisation pour une application client-serveur simple. Serveur multiprogrammé (1 nouveau processus par client servi), avec deux méthodes : création dynamique et pool de processus. Ce travail sera prolongé par une APNEE (projet non encadré, évalué)</p>
		TP6 (1h30)	<p>Utilisation des <i>sockets</i> Unix pour la programmation d'un mini-serveur web ; ce TP se prête à diverses extensions, dont certaines seront proposées dans une APNEE.</p>