

Systemes et Re-seaux – MIAGE 2

Systemes d'Exploitation

Introduction aux TP

Sara Bouchenak

Sara.Bouchenak@imag.fr
<http://sardes.inrialpes.fr/~bouchena/teaching>



Librairie d'utilitaires

- Pour la réalisation des TP de systemes d'exploitation, une librairie de fonctionnalites est fournie :
 - csapp.h
 - csapp.c
- Disponible sur la page web du cours



Interet de la librairie

- Simplifier l'utilisation de primitives Unix
- En particulier, la librairie prend en charge la gestion des cas d'erreurs des primitives
- La librairie fournit de nouvelles versions des primitives Unix, qui gèrent les cas d'erreurs



Exemple : Sans utilisation de la librairie

```
pid_t pid;

if ((pid = fork()) < 0) {
    fprintf(stderr, "%s: %s\n", msg, strerror(errno));
    exit(0);
}
// Utilisation du pid
...

if ((pid = fork()) < 0) {
    fprintf(stderr, "%s: %s\n", msg, strerror(errno));
    exit(0);
}
// Utilisation du pid
...
```

code long et redondant



Exemple : Avec utilisation de la librairie



```
pid_t pid;
pid = Fork();
// Utilisation du pid
...
pid = Fork();
// Utilisation du pid
...
```

code plus simple

Exemple : Mise en œuvre de Fork dans la librairie csapp.c



```
pid_t Fork(void)
{
    pid_t pid;

    if ((pid = fork()) < 0)
        unix_error("Fork error");
    return pid;
}

void unix_error(char *msg) /* unix-style error */
{
    fprintf(stderr, "%s: %s\n", msg, strerror(errno));
    exit(0);
}
```

Autres primitives fournies par la librairie



```
/* Process control wrappers */
pid_t Fork(void);
void Execve(const char *filename, char *const argv[], char *const envp[]);
pid_t Wait(int *status);
pid_t Waitpid(pid_t pid, int *iptr, int options);
```

```
/* Signal wrappers */
handler_t *Signal(int signum, handler_t *handler);
```

Résumé



- Consulter les fichiers :
 - csapp.h
 - csapp.c
- Inclure cette librairie aux programmes de vos TP :
 - #include "csapp.h"
- Utiliser, dans les programmes de vos TP, les primitives fournies par librairie