SO-Chuleta.pdf

Anónimo

Sistemas Operativos

2º Grado en Ingeniería Informática

Facultad de Informática de Barcelona (FIB)
Universidad Politécnica de Catalunya

Includes i main:

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
int main(int argc, char *argv[]) {
}
```

Comandos utils:

Veure tots els procesos: ps -e

Funcions:

- waitpid(-1, NULL, 0); // va retornant el pid dels fills quan moren i retorna 0 quan ja no en queden

-execlp("./executable o comand", "argv[0] que es el nom de executable", "argv[1], .. , (char *) 0);

Saber exit code de sortida:

-int exit_code = WEXITSTATUS(status);

Esquema sequencial:

```
_____for (i = 0; i < n_procesos; ++i) {
        ret = fork();
        switch(ret) {
        case -1: perror("Error creando hijo");
                exit(1);
        case 0:
                //codigohijo()
                exit(0);
        }
        waitpid(-1, NULL, 0);
}
```

Esquema concurrente:

```
for (i = 0; i < n_procesos; ++i) {
        ret = fork();
        switch(ret) {
        case -1: perror("Error creando hijo");
                exit(1);
        case 0:
                //codigohijo()
                exit(0);
        }
}
_____while((ret = waitpid(-1, NULL, 0)) > 0);
```

Espera bloqueante:

//Fora del main() "Com a variable global" poso el recibido

```
int recibido = 0;
```

```
_____struct sigaction trat;
_____sigset_t mask;

//Primer bloquejo el senyal que vull esperar per no detectar-lo fins que em posi en sigsuspend
_____sigemptyset(&mask);
      sigaddset(&mask, SIGUSR1);
      sigprocmask(SIG_BLOCK, &mask, NULL);

//Ara modifico la mascara per posar-la en el trat per tal de nomes poder detectar el senyal a
//esperar quan esigui en sigsuspend
      sigfillset(&mask);
      sigdelset(&mask, SIGUSR1);
      trat.sa_flags = 0;
      trat.sa_handler = tratamiento;
      trat.sa_mask = mask;
      sigaction(SIGUSR1, &trat, NULL);

      //Per ultim em poso en sigsuspend:
      sigsuspend(&mask);
```

Espera activa:

```
_____struct sigaction trat;
_____sigset_t mask;

      //Bloquejo tots els senyals menys el que vull rebre
      sigfillset(&mask);
      sigdelset(&mask, SIGALRM);
      sigprocmask(SIG_BLOCK, &mask, NULL);

      //Configuro com tractar el senyal
      trat.sa_flags = 0;
      trat.sa_handler = tratamiento;
      trat.sa_mask = mask;
      sigaction(SIGALRM, &trat, NULL);

      //Faig espera activa
      while(!recibido);
      recibido = 0;
```

Usage:

```
_____void usage() {
            char buff[80];
            sprintf(buff, "Usage: ./programa paramatre1 paramatre2 paramatreN);
            write(1, buff, strlen(buff));
            exit(1);
      }
```

Makefile:

```
all: nom1 nom2 nom3 nomN
nom1:
      gcc -o nom1 nom1.c
nom2:
```

```
	gcc -o nom2 nom2.c
nom3:
	gcc -o nom3 nom3.c
nomN:
	gcc -o nomN nomN.c

clean:
	rm	nom1 nom2 nom3
```