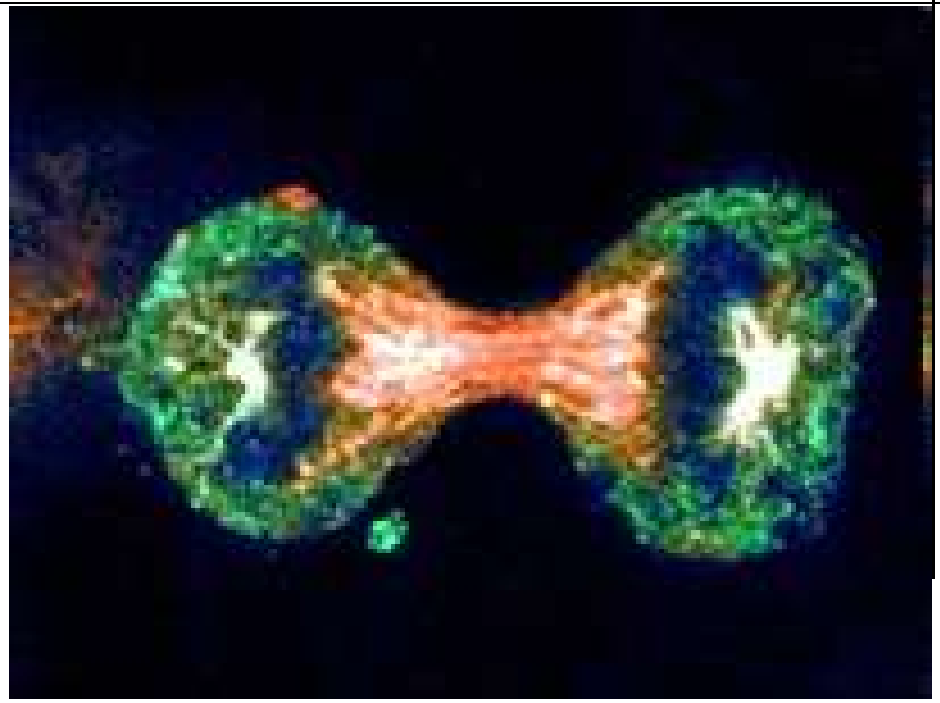
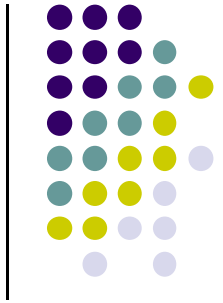
Criação de Processos

Fork( )
Exec( )



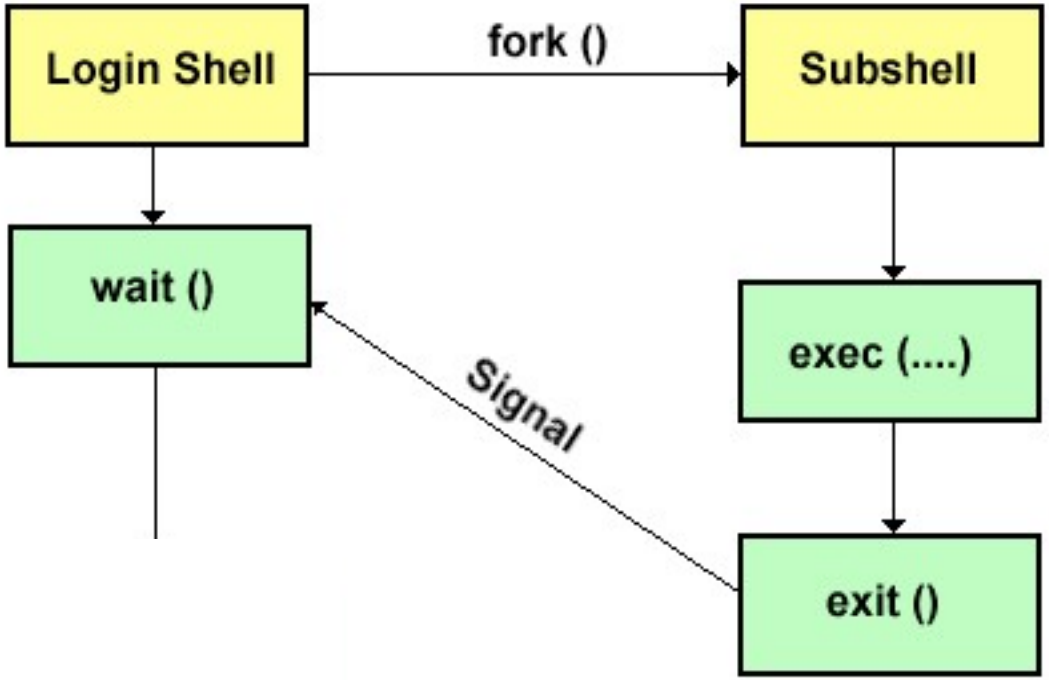
Chamadas de Sistema: Process Management



Gerenciamento de processos

Chamada	Descrição
pid = fork( )	Crie um processo filho idêntico ao processo pai
pid = waitpid(pid, &statloc, options)	Aguarde um processo filho terminar
s = execve(name, argv, environp)	Substitua o espaço de endereçamento do processo
exit(status)	Termine a execução do processo e retorne o estado

Chamada fork( ) / exec( )



Esboço de uma shell

```
while (TRUE) {                                  /* repeat forever */
    type_prompt( );                             /* display prompt */
    read_command (command, parameters)          /* input from terminal */

if (fork() != 0) {                              /* fork off child process */
    /* Parent code */
    waitpid( -1, &status, 0);                   /* wait for child to exit */
} else {
    /* Child code */
    execve (command, parameters, 0);            /* execute command */
 }
}
```

Perguntas?

Exercícios

1) Faça um programa em que três processos executam em paralelo as seguintes ações:

Pai - Imprime os números de 0 a 99, com um intervalo de 1 segundo entre a impressão de cada número. Após imprimir todos os números, imprime a frase “Processo pai vai finalizar” e finaliza quando o filho terminar.

Filho - Imprime os números de 100 a 199, com um intervalo de 2 segundo entre a impressão cada número. Antes de imprimir os números, imprime a frase “Filho foi criado”.

Após imprimir todos os números, imprime a frase “processo filho vai finalizar” e finaliza quando o neto terminar.

Neto - filho do processo Filho (ou seja, neto do processo Pai). Imprime os números de 200 a 299, com um intervalo de 3 segundos entre cada número. Antes de imprimir os números, imprime a frase “Neto foi criado”.

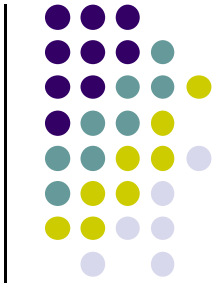
Após imprimir todos os números, imprime a frase “processo neto vai finalizar” e finaliza o processo.

Importante: Em cada printf os processos devem imprimir o seu pid e o pid do seu pai.

DICA: A chamada ao sistema sleep(1) bloqueia o processo por 1 segundo.

Pergunta-se: É possível observar os processos executando em paralelo? Como?

Exercícios



2) Implemente os seguintes programas: o primeiro exibe a mensagem “alo mundo!”, o segundo implementa o programa echo do Unix, que exibe no terminal os argumentos do programa. Compile estes programas. Elabore um programa que crie e execute os dois programas que você escreveu: o alomundo e o echo. Utilize alguma função da família "execv" para realizar esta atividade.

DICA: Para saber os protótipos das funções disponíveis execute o comando “man” no Terminal (para acessar o manual: "man execv").

Pergunta1: O que você observou em termos de semelhanças e diferenças para executar o alomundo e o echo?

Pergunta2: Indique como você decidiu implementar os programas. Há concorrência? Há hierarquia entre os processos? Explique

Leia o texto no site sobre como os laboratórios devem ser entregues.