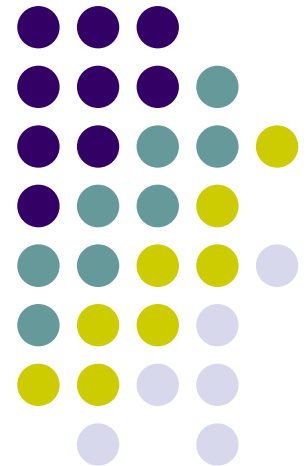
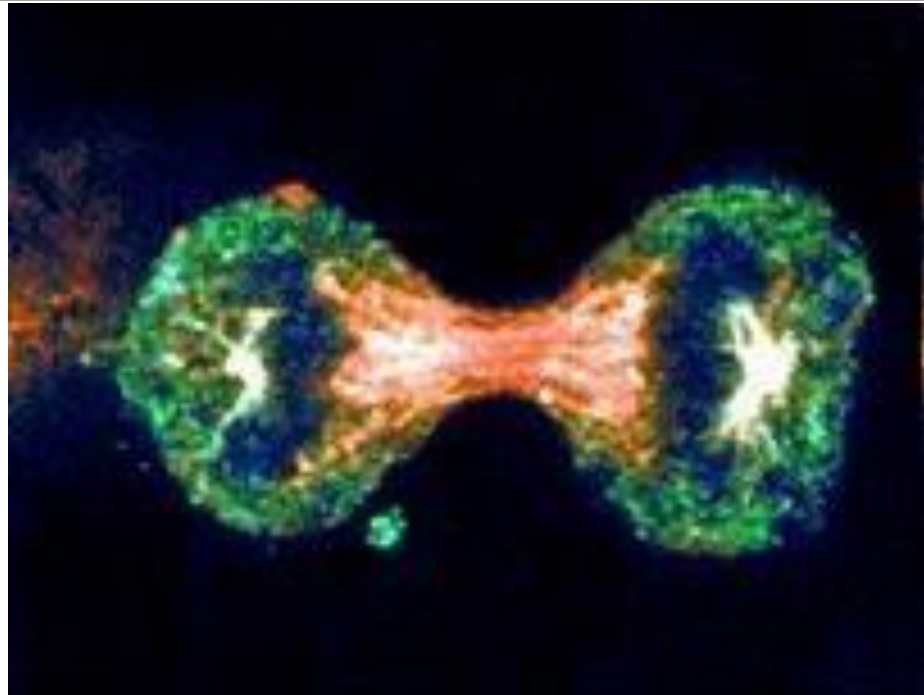
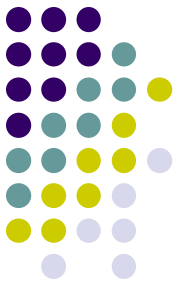


Criação de Processos

Fork()
Exec()

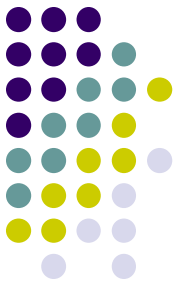




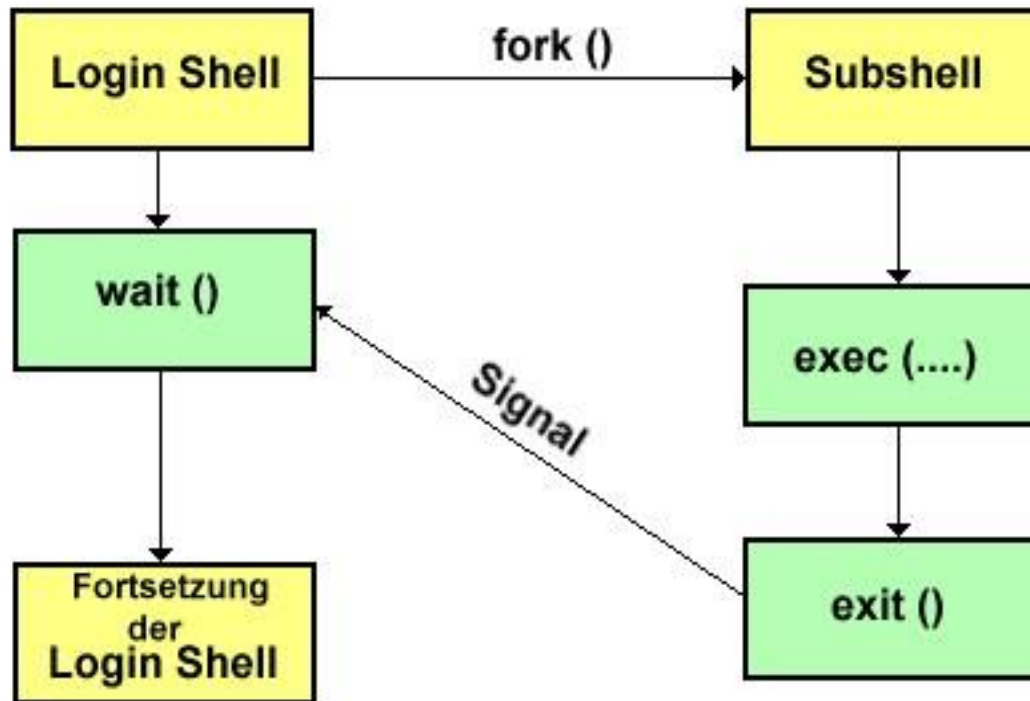
Chamadas de Sistema: Process Management

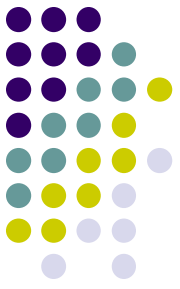
Gerenciamento de processos

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



Chamada fork() / exec()



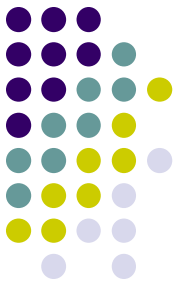


Esboço de uma shell

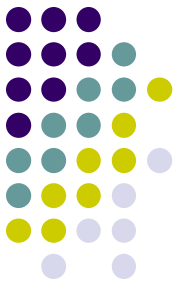
```
while (TRUE) {                                /* repeat forever */
    type_prompt( );                            /* display prompt */
    read_command (command, parameters)        /* input from terminal */

    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

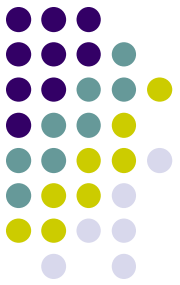


1) Utilizando os comandos **fork()**, **waitpid()** e **exit()** faça um programa composto por dois processos: processo pai e processo filho.

Pai: Imprime “*Processo pai iniciado*”, o seu pid e o pid de seu filho, espera o filho terminar e imprime “*Processo pai finalizado*”.

Filho: Imprime “*Processo filho iniciado*”, o seu pid, o pid do seu pai, e ao terminar imprime “*Processo filho finalizado*”.

DICA: Busque na internet o comando utilizado para obter o PID (process id) de um processo.



Exercícios

2) Ainda com um programa composto por dois processos, crie uma variável visível ao pai e ao filho iniciada com o valor zero.

O **pai** deve executar um loop de 100 vezes, somar 5 a esta variável a cada iteração do loop e imprimir o texto:

Pai <PID> - <valor da variável>.

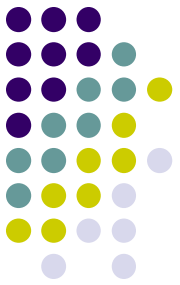
O **filho** deve executar um loop de 100 vezes, somar 10 a esta variável a cada iteração do loop e imprimir o texto:

Filho <PID> - <valor da variável>.

- a) O que você observou sobre o valor da variável no pai e no filho?
- b) O que você observou sobre a concorrência entre os processos?
- c) Explique o por quê destes comportamentos.

DICA: Para saber os protótipos das funções `fork` e `waitpid` execute o comando “man” no Terminal para acessar o manual: “man fork” e “man waitpid”.

Exercícios



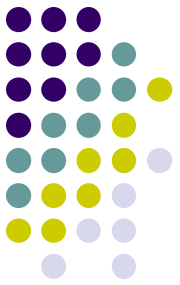
3) Faça um programa em que três processos executam em paralelo as seguintes ações:

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.



Exercícios

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

- a) É possível observar os processos executando em paralelo?
- b) Que alterações devem ser feitas em seu programa para que primeiro sejam exibidas as impressões do neto, depois do filho e depois do pai?
- 4) Crie um programa que realize a impressão da frase: "Alo mundo", no arquivo "alomundo.c". Compile este programa. Em seguida, crie um programa que execute o programa alomundo criado por você. 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**").

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