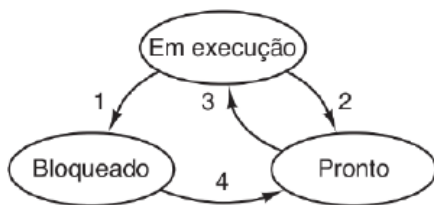




Lista de Exercícios 2 - Processos

- 1) Qual a relação entre programa e processo?
- 2) Cite os 4 eventos principais de criação de novos processos em um sistema computacional.
- 3) Cite 4 razões principais para o término de um processo. Explique cada uma delas.
- 4) O diagrama abaixo ilustra os estados básicos de um processo durante seu tempo de vida.



Responda:

- a) O que significa cada um destes estados?
 - b) O que significa cada uma destas transições?
 - c) Por que algumas transições não são possíveis?
- 5) Para cada processo, o sistema operacional mantém um registro contendo diversas informações acerca do processo. Cite estas informações.
 - 6) Descreva o funcionamento das chamadas de sistema **fork()**, **exec()**, **wait()** e **exit()**.
 - 7) Incluindo o processo pai inicial, quantos processos são criados pelo programa abaixo?

```
#include <stdio.h>
#include <unistd.h>
```

```
int main(){
    fork();
    fork();
    fork();
    return 0;
}
```

- 8) Usando o programa abaixo, identifique os valores escritos nas linhas **A**, **B**, **C** e **D**. (Suponha os pids reais do pai e do filho sejam 2600 e 2603, respectivamente.)

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main(){
    pid_t pid, pid1;

    pid = fork();

    if(pid < 0){
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if(pid == 0){
        pid1 = getpid();
        printf("child: pid = %d",pid); /*A*/
        printf("child: pid1 = %d",pid1); /*B*/
    }
    else{
        pid1 = getpid();
        printf("parent: pid = %d",pid); /*C*/
        printf("parent: pid1 = %d",pid1); /*D*/
    }
    return 0;
}
```

- 9) Trocar um processo por outro no processador demanda tempo de execução do sistema operacional. Suponha que um SO permite que um processo utilize o processador por um tempo máximo de 200 ms. Suponha também que ele leve 15 ms para trocar um processo por outro na execução e que os processos em execução na máquina não fazem Entrada/Saída (Limitados pela CPU). Responda:

- a) Do tempo total de uso desse computador, qual a taxa de tempo gasta em trocas?
- b) Se duplicarmos o tempo máximo que um processo pode usar o processador, qual a nova taxa de tempo gasta em trocas?
- c) Qual o problema de aumentarmos demais o tempo máximo de uso do processador por processo?

- 10) Qual a diferença entre thread e processo?

- 11) Por que um processo iria querer possuir várias threads?

- 12) O que possui cada thread de um processo?

- 13) Qual a principal diferença entre thread de usuário e thread de núcleo?

14) Cite vantagens e desvantagens de thread de usuário e de núcleo.

15) Como funcionam as implementações híbridas de threads?

16) A criação e término de threads são parecidos com o de processos. A API Pthreads possui algumas chamadas de função responsáveis por isso. Descreva a funcionalidade de cada uma das funções Pthreads abaixo:

- a) pthread_create
- b) pthread_exit
- c) pthread_join
- d) pthread_yield

17) O programa abaixo usa a API Pthreads. Qual seria a saída do programa na linha **C** e **P**.

```
#include <pthread.h>
#include <stdio.h>

int value = 0;
void *runner(void *param){
    value = 5;
    pthread_exit(0);
}

int main(int argc, char *argv[]){
    int pid;
    pthread_t tid;

    pid = fork();

    if(pid == 0){
        pthread_create(&tid, NULL, runner, NULL);
        pthread_join(tid, NULL);
        printf("child: value=%d", value); /*C*/
    }
    else if(pid > 0){
        wait(NULL);
        printf("parent: value=%d", value); /*P*/
    }
}
```