

Disciplina: Sistema Operacionais Embarcados **Código:** 120961 **Turma:** A

Professor: Diogo Caetano Garcia

Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 06_Processos_2

1. Crie um código em C para gerar três processos-filho usando o fork().

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

int main(int argc, char **argv){
    pid_t child_pid_1;
    pid_t child_pid_2;
    pid_t child_pid_3;

    printf("numero do processo original: %d\n", (int) getpid());

    child_pid_1 = fork();

    if(child_pid_1 == 0)
    {
        printf("filho_1 %d\n", (int) getpid());
    }

    else
    {
        child_pid_2 = fork();
        if(child_pid_2 == 0)
        {
            printf("filho_2 %d\n", (int) getpid());
        }

        else
        {
            child_pid_3 = fork();
            if(child_pid_3 == 0)
            {
                printf("filho_3 %d\n", (int) getpid());
            }
        }
    }
}
```

```
    else
    {
        printf("pai %d\n", (int) getpid());
    }
}
}
return 0;
}
```

2. Crie um código em C que recebe o nome de diversos comandos pelos argumentos de entrada (argc e *argv[]), e executa cada um sequencialmente usando system(). Por exemplo, considerando que o código criado recebeu o nome de 'serial_system', e que ele foi executado na pasta '/Sistemas_Embarcados/Code/06_Processos', que contem diversos arquivos:

```
$ ./serial_system pwd echo ls echo cal
$ ~/Sistemas_Embarcados/Code/06_Processos
$
$ Ex1.c  Ex2b.c  Ex4.c  Ex6.c
$ Ex2a.c  Ex3.c  Ex5.c  serial_system
$
$  Março 2017
$ Do Se Te Qu Qu Se Sá
$      1  2  3  4
$ 5  6  7  8  9 10 11
$ 12 13 14 15 16 17 18
$ 19 20 21 22 23 24 25
$ 26 27 28 29 30 31
```

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

```
int main(int argc, char **argv)
{

    char *argumentos;
    int i;

    for(i=1;i<=argc;i++)
    {
        argumentos = argv[i];
        system(argumentos);
    }
}
```

```
}  
return 0;  
}
```

3. Crie um código em C que recebe o nome de diversos comandos pelos argumentos de entrada (argc e *argv[]), e executa cada um usando fork() e exec().

```
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <string.h>  
  
int main(int argc, const char *argv[]){  
    int i;  
    char *user_input[2]={NULL,NULL};  
    for (i=1; i<argc; i++){  
        if (fork() == 0){  
            user_input[0]=(char*)argv[i];  
            execvp(user_input[0], user_input);  
            exit(0);  
        }  
    }  
    wait(NULL);  
}
```

4. Crie um código em C que gera três processos-filho usando o fork(), e que cada processo-filho chama a seguinte função uma vez:

```
#include <stdio.h>  
#include <fcntl.h>  
#include <stdlib.h>  
#include <unistd.h>  
#include <string.h>  
#include <sys/types.h>  
int v_global = 0;  
void Incrementa_Variavel_Global(pid_t id_atual)  
{  
    v_global++;  
    printf("ID do processo que executou esta funcao = %d\n", id_atual);  
    printf("v_global = %d\n", v_global);  
}  
int main(int argc, char const *argv[])  
{  
    pid_t child_pid_1;  
    pid_t child_pid_2;  
    pid_t child_pid_3;  
    printf("%d\n", (int) getpid());
```

```
child_pid_1 = fork();
if (child_pid_1==0)
{
    printf("filho1\n");
    Incrementa_Variavel_Global(getpid());
}
else
{
    child_pid_2 = fork();
    if (child_pid_2==0)
    {
        printf("filho2\n");
        Incrementa_Variavel_Global(getpid());
    }
    else
    {
        child_pid_3 = fork();
        if (child_pid_3==0)
        {
            printf("filho3\n");
            Incrementa_Variavel_Global(getpid());
        }
        else
        {
            printf("pai\n");
        }
    }
}
return 0;
}
```

5. Repita a questão anterior, mas desta vez faça com que o processo-pai também chame a função `Incrementa_Variavel_Global()`.

Resposta: a variável global `v_global` foi compartilhada por todos os processos-filho e o processo-pai, ou cada processo enxergou um valor diferente para esta variável?

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

```
int v_global = 0; // Variavel global para este exemplo
void Incrementa_Variavel_Global(pid_t id_atual)
{
    v_global++;
    printf("ID do processo que executou esta funcao = %d\n", id_atual);
}
```

```
printf("v_global = %d\n", v_global);  
}  
  
int main(){  
int kid=0;  
for(kid=0; kid<3; ++kid){  
int pid = fork();  
pid_t id_atual=getpid();  
if (pid<0){  
exit(-1);  
}  
else if(pid==0){  
Incrementa_Variavel_Global(id_atual);  
printf("\nEsse é o processo Luke.\n");  
exit(0);  
}  
else{  
Incrementa_Variavel_Global(id_atual);  
printf("\nEsse é o processo Darth Vader.\n");  
wait(NULL);}}  
}
```