Operációs rendszerek BSc

9. gyakorlat

2021. április 19. (hétfő) 16:00 – 18:00

Készítette:

Szeli Márk

Gazdaságinformatikus

B8VNQ7

1. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetéket, a gyerek processz beleír egy szöveget a csővezetékbe (A kiírt szöveg: XY neptunkod), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre. Mentés: *neptunkod_unnamed.c*

B8VNQ7_unnamed.c:

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

int
main(int argc, char *argv[])
{
    int pipefd[2];
    pid_t cpid;
    char buf;

    if(argc != 2)
    {
        fprintf(stderr, "Használata: %s <string>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    if(pipe(pipefd) == -1)
    {
        perror("PIPE");
        exit(EXIT_FAILURE);
    }

    printf("PID erteke: %d, fd1: %d, fd2: %d\n", getpid(), pipefd[0], pipefd[1]);

    cpid = fork();

    if(cpid == -1)
    {
        perror("FORK");
        exit(EXIT_FAILURE);
    }

    if(cpid == 0)
    {
        close(pipefd[1]);
        while (read(pipefd[0], &buf, 1) > 0)
        {
            write(STDOUT_FILENO, &buf, 1);
        }
        write(STDOUT_FILENO, "\n", 1);
        close(pipefd[0]);
        exit(EXIT_SUCCESS);
    }
```

```
    else
    {
        close(pipefd[0]);
        write(pipefd[1], argv[1], strlen(argv[1]));
        close(pipefd[1]);
        wait(NULL);
        exit(EXIT_SUCCESS);
    }
}
```

Eredmény:

```
mark@mark-K72Jr:~$ ./B8VNQ7_unnamed.out Szeli_Mark_B8VNQ7
PID erteke: 2686, fd1: 3, fd2: 4
Szeli_Mark_B8VNQ7
```

2. Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetéket (neve: neptunkod), a gyerek processz beleír egy szöveget a csővezetékbe (A hallgató neve: pl. Keserű Ottó), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre. Mentés: *neptunkod_named.c*

B8VNQ7_named.c:

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

int main()
{
    pid_t cpid;
    int visszater;
    int fd;
    char buffer[64];
    char B8VNQ7[] = "B8VNQ7";

    visszater = mkfifo(B8VNQ7, 0666);
    cpid = fork();

    if(visszater == -1)
    {
        perror("mkfifo()");
        exit(-1);
    }

    if(cpid == 0)
    {
        printf("Gyermekprocessz PID erteke: %d\n", getpid());
        fd = open(B8VNQ7, O_WRONLY);

        if(fd == -1)
        {
            perror("open()");
            exit(-1);
        }

        strcpy(buffer, "Szeli Mark, B8VNQ7\n");
        write(fd, buffer, strlen(buffer));
        printf("Gyermek PID erteke iras utan: %d\n", getpid());
        close(fd);
    }
```

```
44      else
45      {
46          printf("Szuloprocessz PID erteke: %d\n", getpid());
47          fd = open(B8VNQ7, O_RDONLY);
48          if(fd == -1)
49          {
50              perror("open()");
51              exit(-1);
52          }
53
54          printf("Szuloprocessz PID erteke olvasas elott: %d\n", getpid());
55          visszater = read(fd, &buffer, sizeof(buffer));
56          printf("read() %d byte-ot olvasott be, string: %s\n", visszater, buffer);
57          close(fd);
58          unlink(B8VNQ7);
59      }
60
61      return 0;
62  }
```

Eredmény:

```
mark@mark-K72Jr:~$ ./B8VNQ7_named.out
Szuloprocessz PID erteke: 2784
Gyermekprocessz PID erteke: 2785
Szuloprocessz PID erteke olvasas elott: 2784
read() 19 byte-ot olvasott be, string: Szeli Mark, B8VNQ7
@
Gyermek PID erteke iras utan: 2785
```