

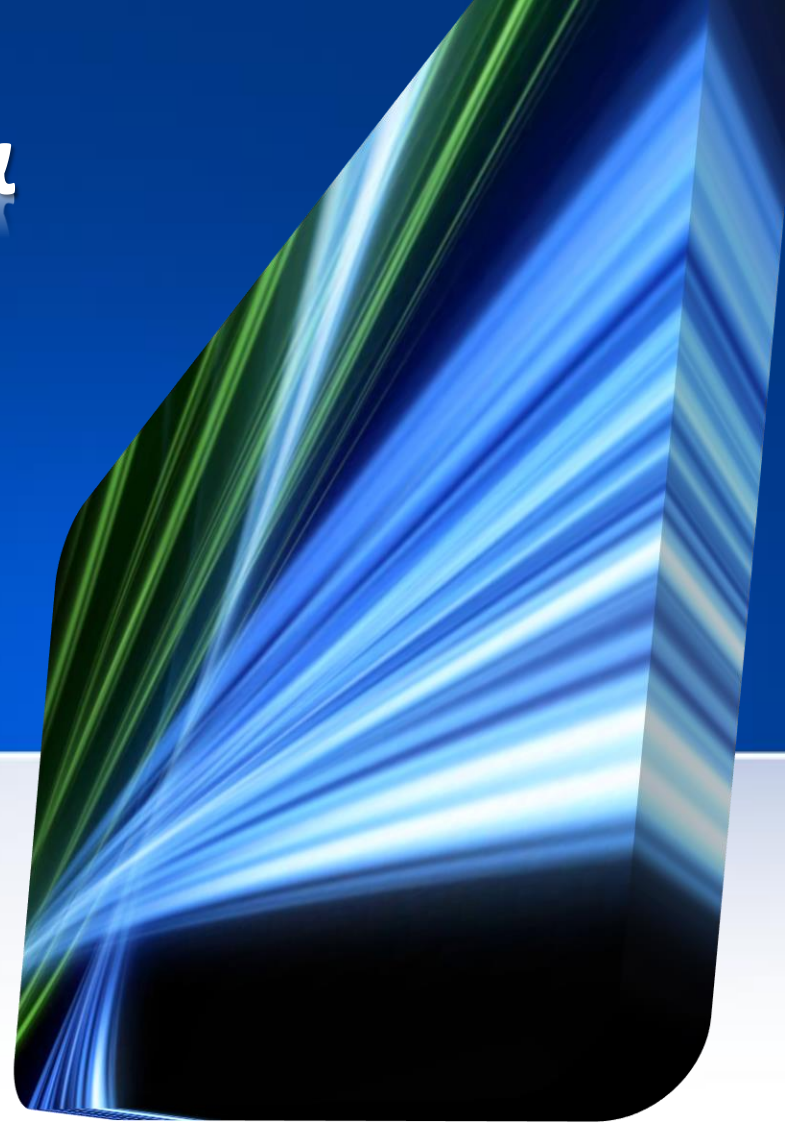
Λειτουργικά Συστήματα

6ο εξάμηνο ΣΗΜΜΥ

Ακ. έτος 2020-2021

Εργαστηριακή Άσκηση 3

Χειρισμός Διοχετεύσεων



Εργαστηριακή Άσκηση 3



Να γραφτεί πρόγραμμα σε γλώσσα προγραμματισμού C και περιβάλλον Linux στο οποίο η διεργασία πατέρας δημιουργεί n διεργασίες παιδιά. Η παράμετρος n ορίζεται από τον χρήστη από την γραμμή εντολών.

Ο χρήστης ορίζει επίσης μια δεύτερη παράμετρο k ο οποίος είναι ένας ακέραιος αριθμός για του οποίου θέλουμε να υπολογίσουμε το παραγοντικό $k!$

Εργαστηριακή Άσκηση 3



Παράδειγμα.

αν το εκτελέσιμο ονομάζεται `assign3`, με την παρακάτω εντολή `./assign3 4 9` ο χρήστης ζητάει με χρήση 4 διεργασιών να υπολογιστεί το 9!

Εργαστηριακή Άσκηση 3



Κάθε διεργασία παιδί πραγματοποιεί ένα πολλαπλασιασμό και προωθεί το αποτέλεσμα στην επόμενη διεργασία έως ότου έχουμε το τελικό αποτέλεσμα.

Η τελευταία διεργασία στέλνει τα αποτελέσματα της στην πρώτη διεργασία.

Η διεργασία πατέρας στέλνει τον αριθμό 1 στην πρώτη διεργασία για να ξεκινήσουν οι υπολογισμοί.

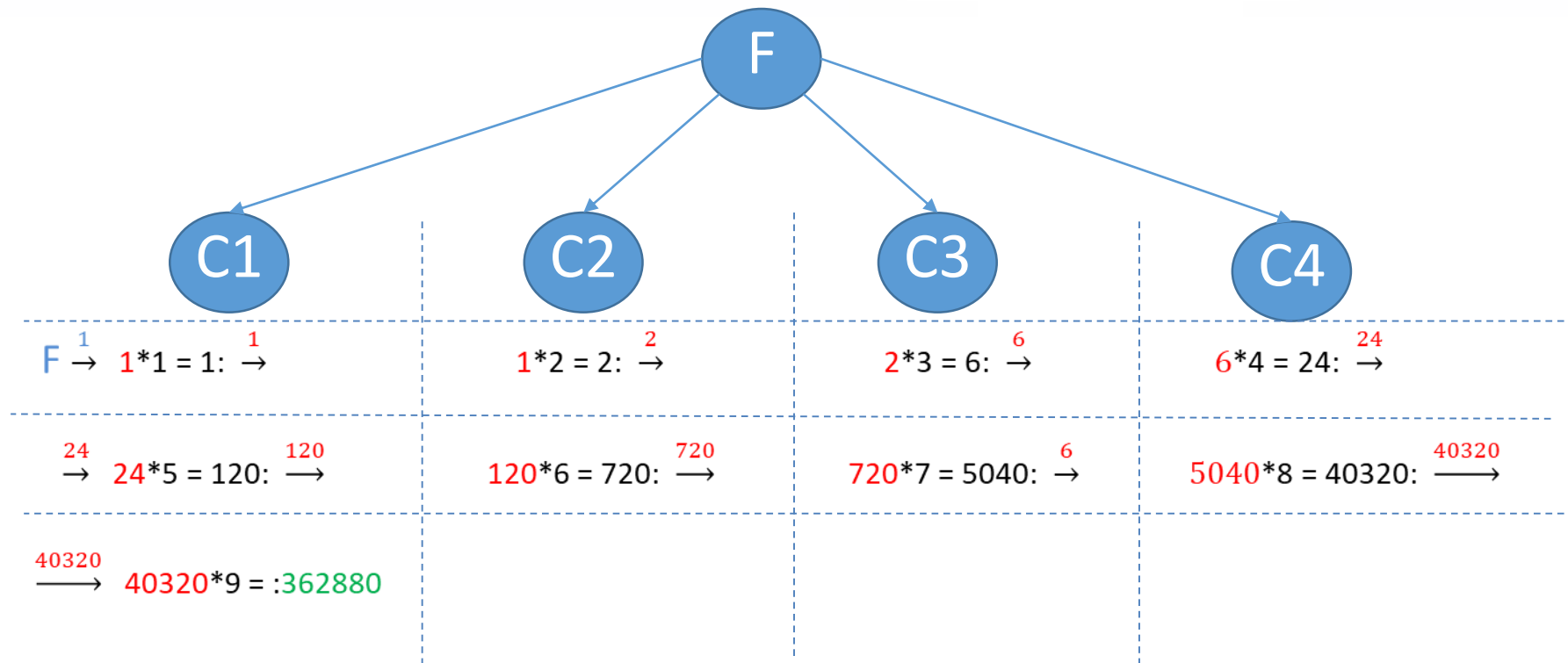
Η διεργασία που πραγματοποίησε τον τελευταίο υπολογισμό εκτυπώνει το αποτέλεσμα

Εργαστηριακή Άσκηση 3



Για το προηγούμενο παράδειγμα

`./assign3 4 9` θα γίνουν οι παρακάτω πράξεις:



Εργαστηριακή Άσκηση 3



Η C1 διαβάζει 1 από τον πατέρα γράφει 1 στην C2

Η C2 διαβάζει 1 από την C1 γράφει 2 στην C3

Η C3 διαβάζει 2 από την C2 γράφει 6 στην C4

Η C4 διαβάζει 6 από την C3 γράφει 24 στην C1

Η C1 διαβάζει 24 από την C4 γράφει 120 στην C2

....

Εργαστηριακή Άσκηση 3



Οι διεργασίες επικοινωνούν και
συγχρονίζονται μέσω διοχετεύσεων (pipes)

Διοχετεύσεις Pipes



- ❑ Λειτουργούν ως σωληνώσεις που επιτρέπουν σε διεργασίες να επικοινωνούν
- ❑ Οι συνηθισμένοι διοχετεύσεις δεν είναι προσβάσιμες έξω της διεργασία που τις δημιουργήσε.
- ❑ Συνήθως, μια γονική διαδικασία δημιουργεί μια διοχέτευση και την χρησιμοποιεί για να επικοινωνεί με μια διεργασία που δημιουργήσε.
- ❑ Επώνυμες διοχετεύσεις (Named pipes) – είναι προσβάσιμες χωρίς να υπάρχει σχέση γονέα-παιδιού.

Κοινές Διοχετεύσεις

Ordinary Pipes



- ❑ Οι κοινές διοχετεύσεις επιτρέπουν την επικοινωνία σε τυπικό τρόπο παραγωγού-καταναλωτή
- ❑ Ο παραγωγός γράφει στο ένα άκρο (άκρο εγγραφής **write-end**)
- ❑ Ο καταναλωτής διαβάζει από το άλλο άκρο (άκρο ανάγνωσης **read-end**)
- ❑ Επομένως, οι κοινές διοχετεύσεις είναι μονής κατεύθυνσης
- ❑ Απαιτείται σχέση μεταξύ των διεργασιών επικοινωνίας
- ❑ Στα Windows καλούνται αυτές οι ανώνυμες διοχετεύσεις **anonymous pipes**

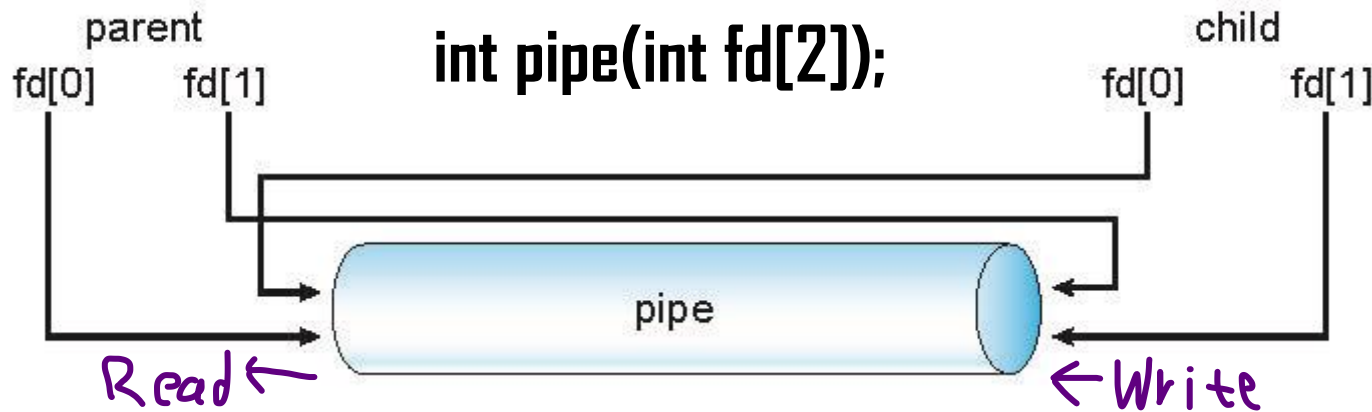
Επώνυμες διοχετεύσεις Named Pipes



- ❑ Οι Επώνυμες διοχετεύσεις παρέχουν ένα ισχυρότερο εργαλείο επικοινωνίας
- ❑ Η επικοινωνία είναι αμφίδρομη
- ❑ Δεν είναι απαραίτητη η σχέση μεταξύ των διεργασιών επικοινωνίας
- ❑ Διάφορες διεργασίες μπορούν να χρησιμοποιήσουν την διοχέτευση για επικοινωνία
- ❑ Παρέχεται σε συστήματα UNIX και Windows

Κοινές Διοχετεύσεις

Ordinary Pipes



Η κλήση συστήματος **pipe()** δημιουργεί μια διοχέτευση, ένα κανάλι δεδομένων μονής κατεύθυνσης που μπορεί να χρησιμοποιηθεί για επικοινωνία μεταξύ διεργασιών. Ο πίνακας `fd` χρησιμοποιείται για την επιστροφή δύο περιγραφητών αρχείων που αναφέρονται στα άκρα της διοχέτευσης. Το στοιχείο `fd[0]` του πίνακα αναφέρεται στο άκρο ανάγνωσης της διοχέτευσης και το `fd[1]` αναφέρεται στο άκρο εγγραφής.

Περισσότερες πληροφορίες:
<https://linux.die.net/man/2/pipe>

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main(void)
{
```

```
    int pd[2], nbytes;
    pid_t childpid;
    char string[] = "Hello, world!\n";
    char readbuffer[80];
    pipe(pd);
```

```
    if((childpid = fork()) == -1) {
        perror("fork");
        exit(1);
    }
```

```
    if(childpid == 0) {
        close(pd[0]);
        write(pd[1], string, (strlen(string)+1));
        exit(0);
```

↗ '/0'

```
    }
    else {
        close(pd[1]);
        nbytes = read(pd[0], readbuffer, sizeof(readbuffer));
        printf("Received string: %s", readbuffer);
    }
    return(0);
}
```

Παράδειγμα Διοχετεύσεων



Χρήσιμα links



http://www.gnu.org/software/libc/manual/html_node/Pipes-and-FIFOs.html#Pipes-and-FIFOs