**PRAKTIKUM SISTEM OPERASI MODUL 8**



**Disusun Oleh :**

**RAMA AZIZ WIBOWO**

**L200210263KELAS E**

**TEKNIK INFORMATIKA**

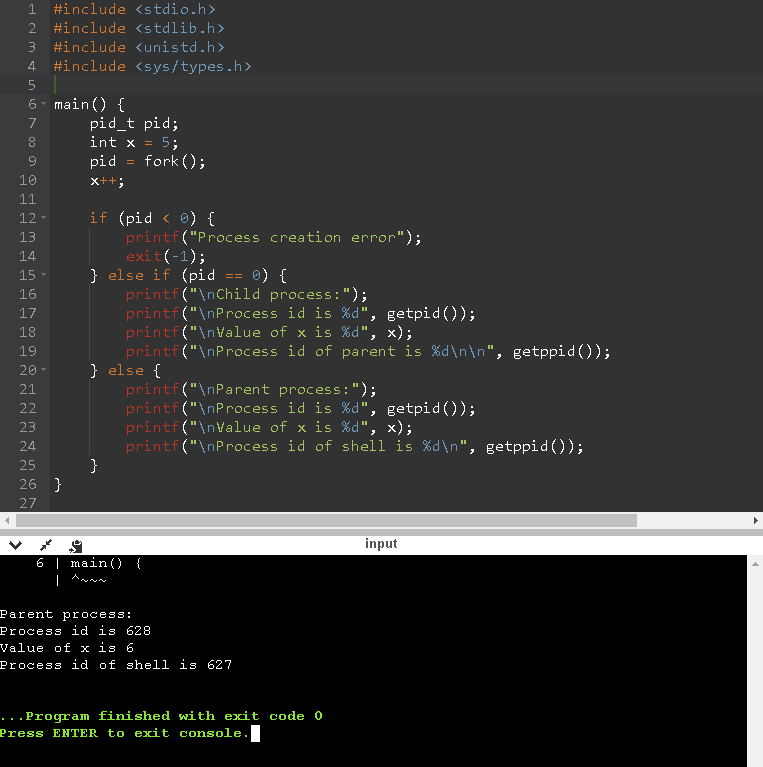
**FAKULTAS KOMUNIKASI DAN INFORMATIKA UNIVERSITAS MUHAMMADIYAH SURAKARTA 2021/2022**

**KEGIATAN PRAKTIKUM:**

1. Membuat sebuah ‘child process’ (proses baru) dengan menggunakan system call ‘fork’. Mambuat program dengan algoritma sebagai berikut :

(contoh program diberikan pada bagian berikutnya).

* 1. Deklarasi sebuah variabel x yang akan diakses bersama antara child proses dan parent proses.
  2. Membuat sebuah child proses menggunakan system call fork.
  3. Jika return value bernilai -1, tampilkan teks ‘Pembuatan proses GAGAL’, dilanjutkan dengan keluar program dengan perintah system call ‘exit’.
  4. Jika return value sama dengan 0 (NOL), Tampilkan teks ‘Child Process’, tampilkanID proses dari child proses menggunakan perintah system call ‘getpid’, tampilkan nilai x, dan tampilkan ID proses parent dengan perintah system call ‘getppid’.
  5. Untuk nilai return value yang lainnya, tampilkan teks ‘Parent process’, tampilkan ID dari parent proses menggunakan perintah system call getpid, tampilkan nilai x, dan tampilkan ID dari proses shell menggunakan perintah system call getppid.
  6. Stop
  7. Kode Pogram & Output:



1. Menghentikan sementara (block) proses parent sampai dengan proses child

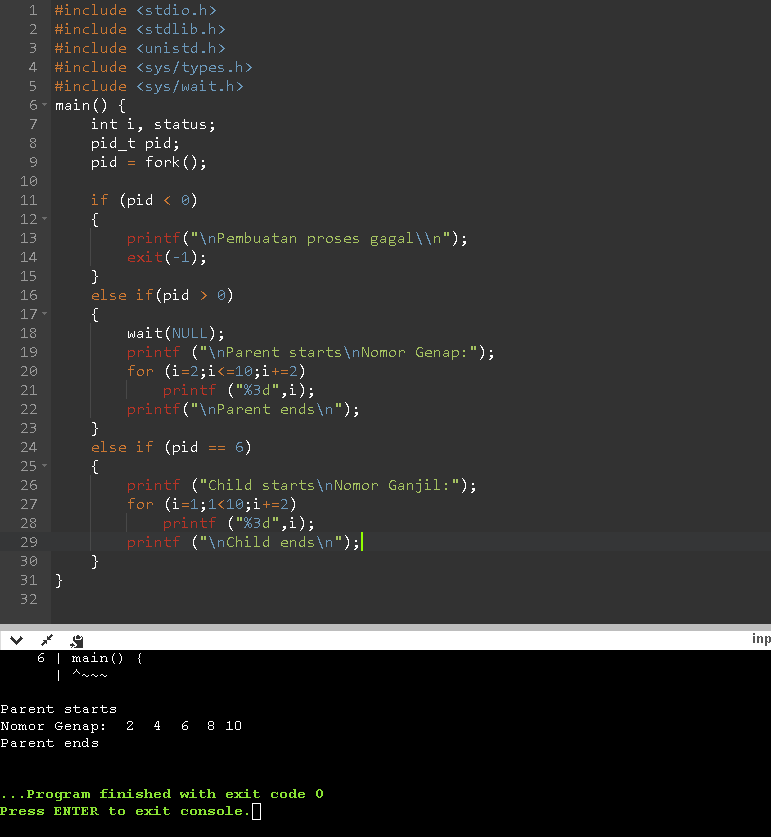
selesai,menggunkaan perintah system call ‘wait’.

Membuat program dengan algoritma sebagai berikut,contoh program diberikan pada bagian berikutnya.

* 1. Membuat sebuah child proses menggunakan sytem call ‘fork’.
  2. Jika return value bernilai -1, selanjutnya tampilkan teks ‘pembuatan proses gagal’, dan kelaur program dengan menggunakan perintah system call ‘exit’.
  3. Jika return value berupa angka positif ( > 0), ‘pause’ hentikan sementara ‘parent’ proses tunggu sampai child proses berakhir dengan menggunakan perintah system call ‘wait’. Tampilkan teks ‘Parent starts’, selanjutnya tampilkan nomor genap mulai dari 0 s/d 10, terakhir tampilkan teks ‘Parent end’.
  4. Jika return value bernilai 0 (NOL), tampilkan teks ‘Child start’, tampilkan nomor ganjil

mulai dari 0 s/d 10, selanjutnya tampilkan teks ‘child ends’

* 1. Stop
  2. Kode Program& Output:



1. Loading program yang dapat dieksekusi dalam sebuah ‘child’ proses menggunakan perintah system call ‘exec’. Membuat program dengan algoritma sebagai berikut:

(contoh program diberikan pada bagian berikutnya).

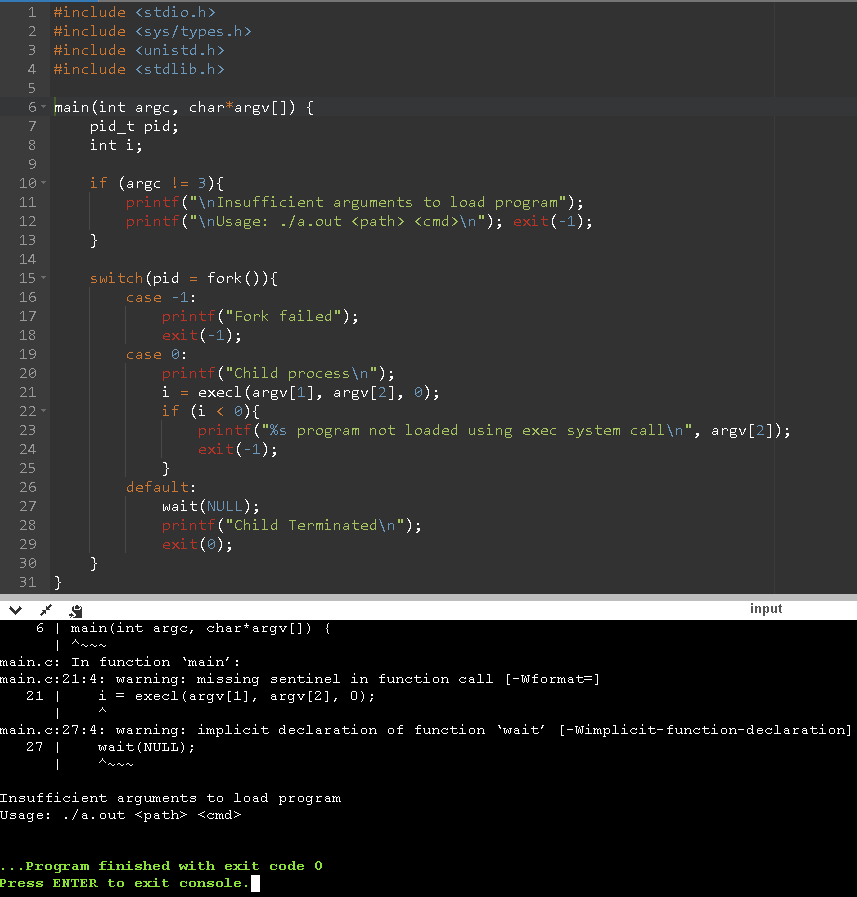
* 1. Jika terdapat 3 argumen dalam command-line berhenti (stop).
  2. Membuat child proses dengan perintah system call ‘fork’
  3. Jika return value adalah -1, selanjuntya tampilkan teks ‘Pembuatan proses Gagal’, dan

keluar program dengan perintah system call exit.

* 1. Jika return value >0 (positif), selanjutnya hentikan parent-proses sementara hingga child-proses berakhir dengan menggunakan perintah system call wait. Tampilkan teks ‘Child berakhir’, dan hentikan parent-proses.
  2. Jika return value sama dengan 0 (NOL), selanjutnya tampilkan teks ‘Child starts’, load program dari lokasi yang diberikan dalam ‘path’ ke dalam child-proses, menggunakan

perintah system call ‘exec’. Jika return value dari perintah ‘exec’ adalah bilangan negatif, tampilkan error yang terjadi dan stop. Hentikan child- proses.

* 1. Stop
  2. Kode Program & Output:



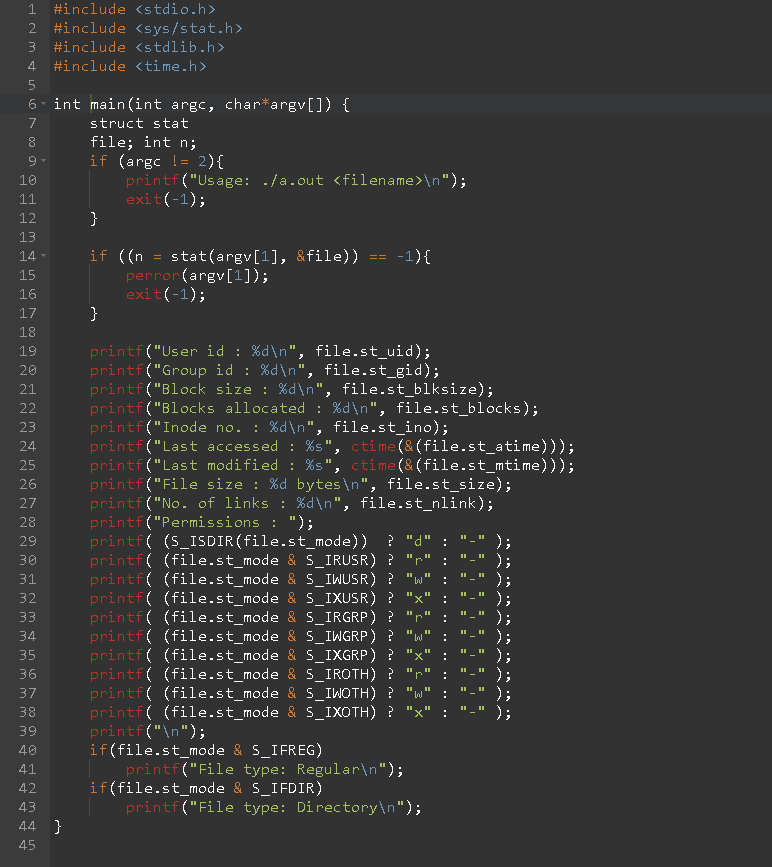
1. Menampilkan status file menggunakan perintah system call ‘stat’

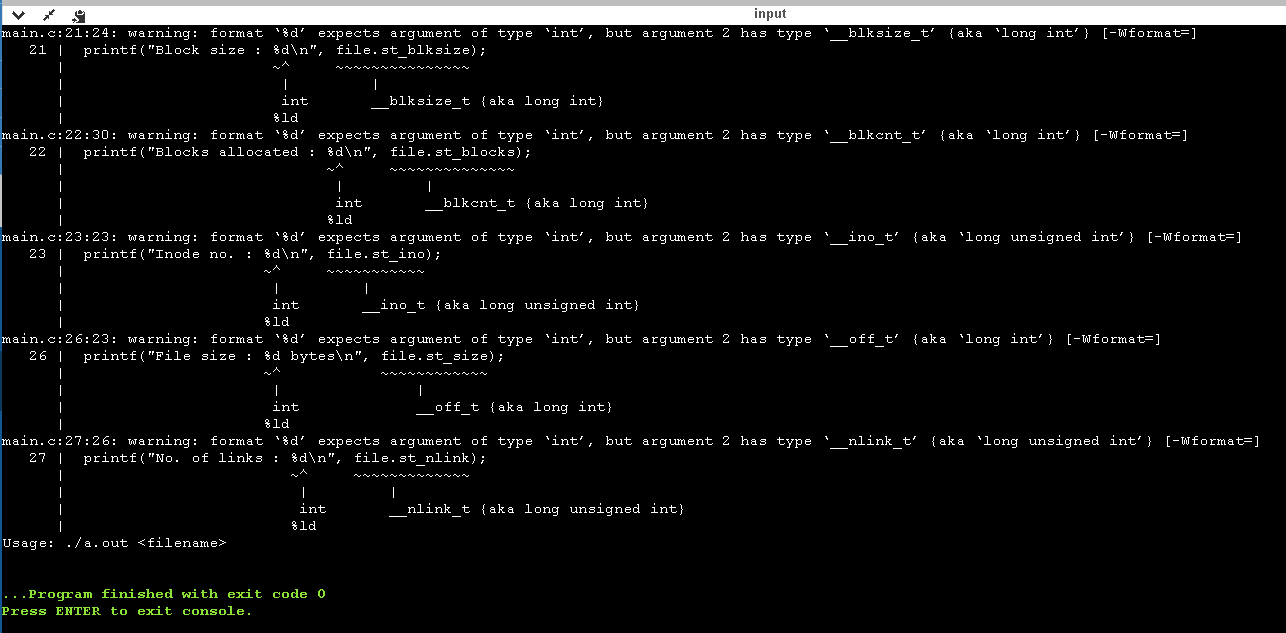
Membuat program dengan algoritma sebagai berikut (contoh code ada di bagian berikutnya):

* 1. Gunakan ‘nama file’ yang diberikan melalui argumen dalam perintah command- line.
  2. Jika ‘nama-file’ tidak ada maka stop disini (keluar program)
  3. Panggil system call ‘stat’ pada ‘nama-file’ tersebut yang akan mengembalikan sebuah

struktur.

* 1. Tampilkan informasi mengenai st\_uid, st\_blksize, st\_block, st\_size, st\_nlink, etc.
  2. Ubah waktu dalam st\_time, st\_mtime dengan menggunakan fungsi ctime.
  3. Bandingkan st\_mode dengan konstanta mode seperti S\_IRUSR, S\_IWGRP, S\_IXOTH dan tampilkan informasi mengenai ‘file-permissions’.
  4. Stop
  5. Kode Program & Output:





1. Menampilkan isi direktori menggunakan perintah system call ‘readdir’

Membuat program dengan algoritma sebagai berikut (contoh code ada di bagian berikutnya):

* 1. Gunakan ‘nama-direktori’ yang diberikan sebagai argumen pada command-line.
  2. Jika direktori tidak ditemukan stop, keluar program
  3. Buka direktori menggunakan perintah system call ‘opendir’ yang akan menghasilkan

sebuah struktur.

* 1. Baca direktori menggunakan perintah system call ‘readdir’ yang juga akan menghasilkan

struktur data.

* 1. Tampilkan d\_name (nama direkrori)
  2. Akhiri pembacaa direktori dengan perintah system call ‘closedir’.
  3. Stop
  4. Kode Program & Output:

