int main(int argc, char** argv)   long i = sysconf();   atoi(char*)
fflush(stdout);

Threads: separate Thread-IDs

int pthread_create(*array, NULL, (void*) methode, (void*) argumente);
errno

int pthread_detach(threadID);

DIR* dir = opendir(char*); if(dir==NULL)

char* path = (char*)calloc(zeichenanzahl, sizeof(zeichen)),
auf NULL prüfen;

errno zuvor auf 0 setzen

struct dirent* dirCont;   while((dirCont=readdir(dir)) != NULL){ }
d_name   d_type   DT_DIR
DT_REG
if(errno) => Fehler, sonst EOF

strcpy(dest, src); memcpy(dest, src); strcat(first, sec);

strstr(where, what); == NULL   int strcmp(s1, s2);
=0 => true !0 => false

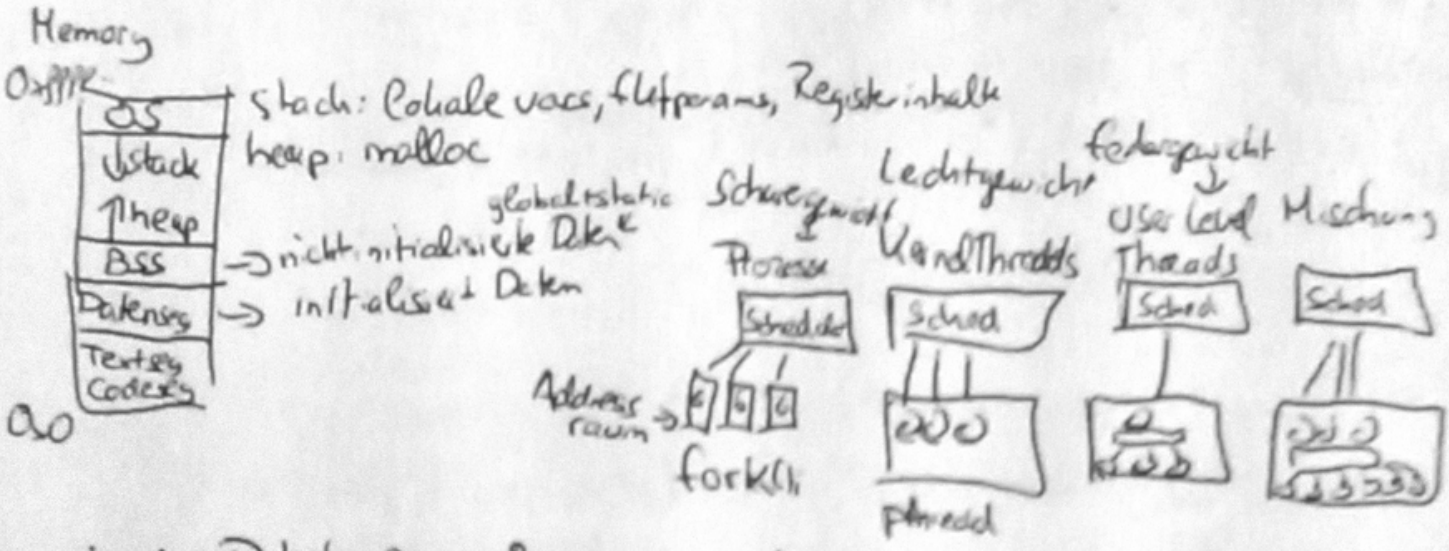
~~Kind~~ Fork:
int pid = fork(); pid=0 => Kindprozess   pid<0 => error, else parent
exit(execvp(char* programm, args));   args => erster param programm, letzter NULL

stat follow symbolic link
lstat not!
lstat(path

struct stat buffer;   if(lstat(path, &buffer)==0
~~if(lstat(path, buffer)) == 0~~   => klappt
buffer.st_size
st_mode => S_ISDIR
S_ISREG

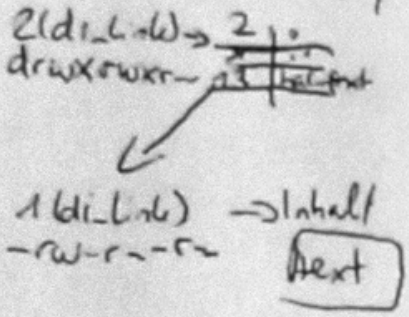
Qsort:
int intCompare(const int*, const int*)
qsort(list*, length list*, sizeof(int*), intCompare);
while
char*(strtok) != NULL   Strtok(string, "_");

fprintf(stderr, str);
errno => perror(String). exit(EXIT_FAILURE).



inode Dateikopf weiß user, group, type, rights, timestamp, #hardlinks, size, addresse auf Speicher

inode number, Zahl, Eindeutig inode id



Adressraum

BS Betriebs
HW Hardware

physikalischer Adressraum (HW)
- nicht jede adresse gültig

Logisch Adressraum (Compiler, Binder, BS)
jede adresse gültig verwendbar

virtueller Adressraum (BS)
logisch ~~mit allen~~ auf alle abgebildet

Trap synchron, vorhersagbar, reproduzierbar,
(z.b. ~~division by zero~~ Syscall (kann beendet werden)

Interrupt
asynchron, unvorhersagbar, nicht reproduzierbar
(z.b. Tastatur)
(darf nicht beendet werden)