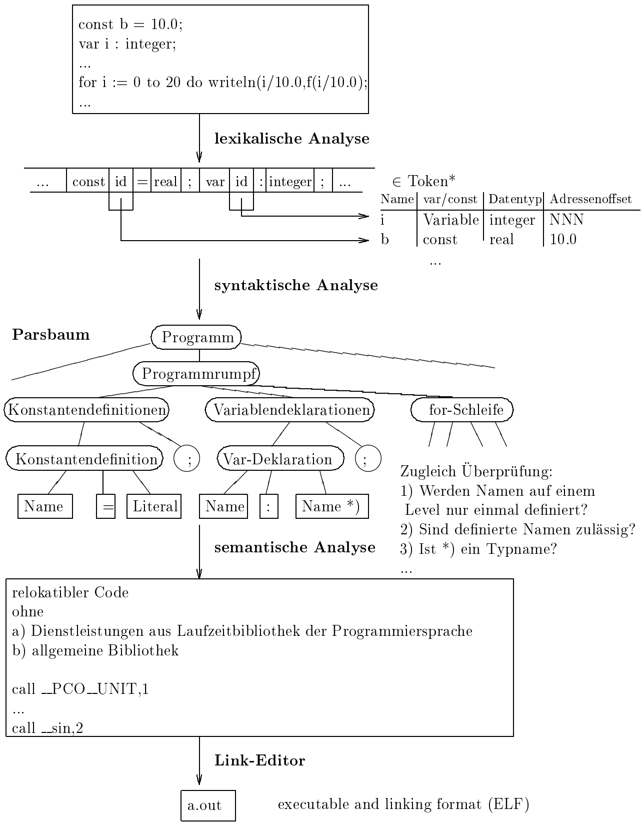
Kapitel 2

Systemsoftware / Basissoftware

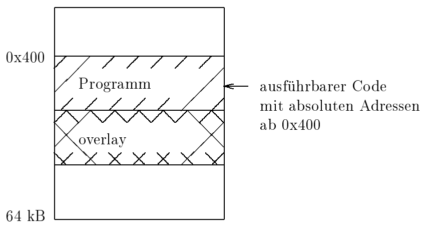
2.1 Compiler und Programmentwicklung

Ein Compiler führt einen in einer höheren Programmiersprache geschriebenen Algorithmus in ein Maschinenprogramm über:



Relokatibler Code:

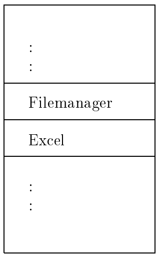
- *CP/M-System:* Falls ein Overlay oder ROM-Code erzeugt werden soll, ist eine neue Compilation mit anderer Startadresse nötig.



- *Idee:* Overlays, Bibliotheksmodule, ... werden *verschiebbar* mit *relativen* Adressen erzeugt.


```
**** RELOCATION INFORMATION ****

.rela.text:

Offset   Symndx  Type  Addend
0x10a9c  63      12    0
0x10aec  63      12    0
0x10ae8  63      9     0
0x10a98  63      9     0
.rela.ex_shared:

Offset   Symndx  Type  Addend
0x215cc  56      3     0
0x215c8  36      3     0
.rela.bss:

Offset   Symndx  Type  Addend
0x21648  46      19    0
0x21a70  53      19    0
.rela.plt:

Offset   Symndx  Type  Addend
0x21558  40      21    0
0x21564  68      21    0
0x21570  41      21    0
0x2157c  55      21    0
0x21588  47      21    0
0x21594  52      21    0
0x215a0  71      21    0
0x215ac  45      21    0
0x215b8  66      21    0
```

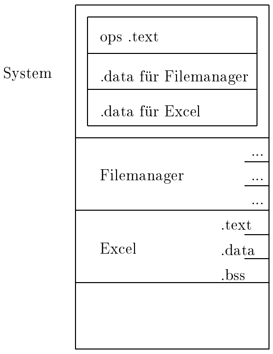
- *Möglichkeit zur Sicherheit von Prozess-Datenbereichen mit Hilfe von relokatiblem Code:*



In Mehrprozesssystemen (preemptiv) sind zur selben Zeit viele Applikationen im Hauptspeicher:

1. relokatibler Code nötig!
2. Falls per Soft-/Hardware die Adressen von Speicherzugriffen validiert werden, so kann der Zugriff auf Datenbereiche anderer Applikationen unterdrückt werden.

Genauer gilt:



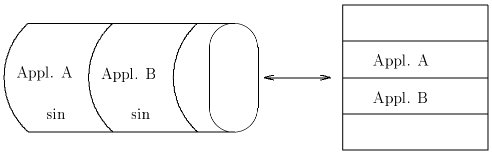
Programm (inkl. Konstanten), `r-x`
initialisierte Daten, `rwx`
nicht initialisierte Daten, `rwx`

```
wminf4% size -f xspread
17(.interp) + 4796 (.hash) + 9536(.dynsym) + 5184(.dynstr)
+ 24(.rela.got) + 48(.rela.plt) + 181176(.text) + 12(.init)
+ 12(.fini) + 13872(.rodata) + 6995(.rodata1) + 72(.got) +
152(.dynamics) + 1360(.plt) + 31676(.data) + 28698(.data1)
+ 14708(.bss) = 299646
wminf%4 size -F xspread
223195(r-x) + 76672(rwx) = 300027
```

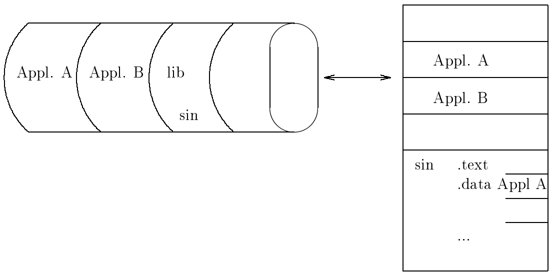


```
wminf%4 size xspread
223195 + 61964 + 14708 = 299867
```

Statisches Linken



dynamisches Linken



`dump -Lv Tabelle` liefert:

```
**** DYNAMIC SECTION INFORMATION ****

               .dynamic:
[INDEX] Tag     Value
[1]     NEEDED  libpc.so.4
[2]     NEEDED  libm.so.1
[3]     NEEDED  libc.so.1
[4]     INIT    0x110ec
[5]     FINI    0x11124
[6]     RPATH   /opt/SUNWspro/lib
```



```
[7]     HASH    0x100e8
[8]     STRTAB  0x10778
[9]     STRSZ   0x225
[10]    SYMTAB  0x102e8
[11]    SYMENT  0x10
[12]    DEBUG   0x0
[13]    TEXTREL0x0
[14]    PLTGOT  0x21528
[15]    PLTSZ   0x6c
[16]    PLTREL  0x7
[17]    JMPREL  0x10a00
[18]    RELA    0x109a0
[19]    RELASZ  0xcc
[20]    RELAENT0xc
```

`nm Tabelle.o` liefert:

```
[Index] Value Size Type Bind Other Shndx Name
[1]     0     0    FILE LOCL 0     ABS   ./Tabelle.p
[2]     0     0    SECT LOCL 0     2
[3]     0     0    SECT LOCL 0     3
[4]     0     0    SECT LOCL 0     4
[5]     0     0    SECT LOCL 0     5
[6]     0     0    SECT LOCL 0     6
[7]     0     0    SECT LOCL 0     7
[8]     4     4    OBJT GLOB 0     COMMON1
[9]     4     4    OBJT GLOB 0     COMMONPROGFRAME
[10]    16    132  FUNC GLOB 0     2     f
[11]    0     0    NOTY GLOB 0     UNDEF __sin
[12]    0     0    NOTY GLOB 0     UNDEF output
[13]    176   396  FUNC GLOB 0     2     program
[14]    0     0    NOTY GLOB 0     UNDEF __PC0__tb_init
[15]    0     0    NOTY GLOB 0     UNDEF __PC0__UNIT
[16]    0     0    NOTY GLOB 0     UNDEF fputc
[17]    0     0    NOTY GLOB 0     UNDEF fprintf
```

`nm hello` zeigt auch die Symbole der eingebundenen Bibliotheksmodule:

```
Symbols from hello:

[Index] Value    Size Type Bind OtherShndx Name
```



```
[1]     0        0    FILE LOCL 0    ABS   hello
[2]     65748    0    SECT LOCL 0    1
[3]     65768    0    SECT LOCL 0    2
[4]     66280    0    SECT LOCL 0    3
[5]     67448    0    SECT LOCL 0    4
[6]     68000    0    SECT LOCL 0    5
[7]     68048    0    SECT LOCL 0    6
[8]     68072    0    SECT LOCL 0    7
[9]     68096    0    SECT LOCL 0    8
[10]    68208    0    SECT LOCL 0    9
[11]    69868    0    SECT LOCL 0    10
[12]    69924    0    SECT LOCL 0    11
[13]    69980    0    SECT LOCL 0    12
[14]    69984    0    SECT LOCL 0    13
[15]    69988    0    SECT LOCL 0    14
[16]    136316   0    SECT LOCL 0    15
[17]    136320   0    SECT LOCL 0    16
[18]    136488   0    SECT LOCL 0    17
[19]    136648   0    SECT LOCL 0    18
[20]    136680   0    SECT LOCL 0    19
[21]    136768   0    SECT LOCL 0    20
[22]    0        0    SECT LOCL 0    21
[23]    0        0    SECT LOCL 0    22
[24]    0        0    SECT LOCL 0    23
[25]    0        0    SECT LOCL 0    24
[26]    0        0    SECT LOCL 0    25
[27]    0        0    SECT LOCL 0    26
[28]    0        0    SECT LOCL 0    27
[29]    0        0    SECT LOCL 0    28
[30]    0        0    FILE LOCL 0    ABS   crti.s
[31]    68208    0    NOTY LOCL 0    9     _ex_text0
[32]    69980    0    NOTY LOCL 0    12    _ex_range0
[33]    136656   0    NOTY LOCL 0    18    _ex_shared0
[34]    0        0    FILE LOCL 0    ABS   crt1.s
[35]    136708   0    NOTY LOCL 0    19    __crt_scratch
[36]    0        0    FILE LOCL 0    ABS   values-Xa.c
[37]    0        0    NOTY LOCL 0    UNDEF
[38]    0        0    NOTY LOCL 0    UNDEF
[39]    0        0    FILE LOCL 0    ABS   ./hello.p
[40]    136768   0    NOTY LOCL 0    20    Bbss.bss
[41]    136720   0    NOTY LOCL 0    19    Ddata.data
[42]    0        0    FILE LOCL 0    ABS   ieee_retro.c
```

```
[43]   136728   35   OBJT LOCL 0    19    sccsid
[44]   0        0    NOTY LOCL 0    UNDEF
[45]   0        0    NOTY LOCL 0    UNDEF
[46]   0        0    FILE LOCL 0    ABS   __ieee_retro.S
[47]   0        0    FILE LOCL 0    ABS   crtn.s
[48]   69868    0    NOTY LOCL 0    9     _ex_text1
[49]   69980    0    NOTY LOCL 0    12    _ex_range1
[50]   136672   0    NOTY LOCL 0    18    _ex_shared1
[51]   68208    244  FUNC GLOB 0    9     _start
[52]   68688    1024 FUNC WEAK 0    9     ieee_retrospective
[53]   136768   4    OBJT GLOB 0    20    i
[54]   136680   4    OBJT GLOB 0    19    _environ
[55]   138160   0    OBJT GLOB 0    20    _end
[56]   137840   320  OBJT WEAK 0    20    _iob
[57]   69796    28   FUNC GLOB 0    9     __get_accrued
[58]   0        0    NOTY WEAK 0    UNDEF _ex_register
[59]   136772   4    OBJT GLOB 0    20    PROGFRAME
[60]   136316   0    OBJT GLOB 0    15    _GLOBAL_OFFSET_TABLE_
[61]   69768    28   FUNC GLOB 0    9     __get_traps
[62]   136536   0    FUNC GLOB 0    UNDEF atexit
[63]   136560   0    FUNC GLOB 0    UNDEF exit
[64]   69824    28   FUNC GLOB 0    9     __get_direction
[65]   68688    1024 FUNC GLOB 0    9     __ieee_retrospective
[66]   69868    56   FUNC GLOB 0    10    _init
[67]   136620   0    FUNC GLOB 0    UNDEF fputc
[68]   136776   1060 OBJT GLOB 0    20    output
[69]   136584   0    FUNC GLOB 0    UNDEF __PC0__tb_init
[70]   136716   4    OBJT GLOB 0    19    ___Argv
[71]   69740    28   FUNC GLOB 0    9     __get_nonstandard
[72]   69852    16   FUNC GLOB 0    9     __get_precision
[73]   136320   0    OBJT GLOB 0    16    _DYNAMIC
[74]   136596   0    FUNC GLOB 0    UNDEF __PC0__UNIT
[75]   137840   320  OBJT GLOB 0    20    __iob
[76]   68480    208  FUNC GLOB 0    9     program
[77]   136572   0    FUNC GLOB 0    UNDEF _exit
[78]   0        0    NOTY WEAK 0    UNDEF _ex_deregister
[79]   136680   4    OBJT WEAK 0    19    environ
[80]   136712   4    OBJT GLOB 0    19    __cg89_used
[81]   136712   4    OBJT GLOB 0    19    __cg92_used
[82]   136712   4    OBJT GLOB 0    19    __fnonstd_used
[83]   136763   0    OBJT GLOB 0    19    _edata
[84]   136488   0    OBJT GLOB 0    17    _PROCEDURE_LINKAGE_TABLE_
```



```
[85]   0        0    NOTY GLOB 0    ABS   __fsr_init_value
[86]   70779    0    OBJT GLOB 0    14    _etext
[87]   69984    4    OBJT GLOB 0    13    _lib_version
[88]   136632   0    FUNC GLOB 0    UNDEF fflush
[89]   69712    28   FUNC WEAK 0    9     ieee_retrospective_
[90]   136548   0    FUNC GLOB 0    UNDEF main
[91]   136684   24   OBJT GLOB 0    19    __environ_lock
[92]   69924    56   FUNC GLOB 0    11    _fini
[93]   136608   0    FUNC GLOB 0    UNDEF fprintf
[94]   69712    28   FUNC GLOB 0    9     __ieee_retrospective_
```

`truss a.out` enthüllt die komplexen Systemprozesse, die ein Programmstart initiiert:

```
execve("./a.out", 0xEFFFF3B0, 0xEFFFF3B8) argc = 1
open("/dev/zero", O_RDONLY) = 3
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE,
3, 0) = 0xEF7C0000
open("/usr/lib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/usr/openwin/lib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/usr/ucblib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/usr/dt/lib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/opt/local/lib/libpc.so.4", O_RDONLY) Err#2 ENOENT
open("/opt/SUNWspro/lib/libpc.so.4", O_RDONLY) = 4
fstat(4, 0xEFFFF064) = 0
mmap(0x00000000, 4096, PROT_READ|PROT_EXEC, MAP_SHARED, 4, 0)
= 0xEF7B0000
mmap(0x00000000, 143360, PROT_READ|PROT_EXEC, MAP_PRIVATE, 4,
0) = 0xEF780000
munmap(0xEF790000, 61440) = 0
mmap(0xEF79F000, 12384, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIX
4, 61440) = 0xEF79F000
close(4) = 0
open("/usr/lib/libm.so.1", O_RDONLY) = 4
fstat(4, 0xEFFFF064) = 0
mmap(0xEF7B0000, 4096, PROT_READ|PROT_EXEC, MAP_SHARED|MAP_FIXED,
4, 0) = 0xEF7B0000
mmap(0x00000000, 155648, PROT_READ|PROT_EXEC, MAP_PRIVATE, 4,
0) = 0xEF750000
munmap(0xEF765000, 61440) = 0
mmap(0xEF774000, 6056, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIXE
4, 81920) = 0xEF774000
close(4) = 0
```



```
open("/usr/lib/libc.so.1", O_RDONLY) = 4
fstat(4, 0xEFFFF064) = 0
mmap(0xEF7B0000, 4096, PROT_READ|PROT_EXEC, MAP_SHARED|MAP_FIXED,
4, 0) = 0xEF7B0000
mmap(0x00000000, 618496, PROT_READ|PROT_EXEC, MAP_PRIVATE, 4,
0) = 0xEF680000
munmap(0xEF6FE000, 61440) = 0
mmap(0xEF70D000, 28768, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIX
4, 512000) = 0xEF70D000
mmap(0xEF715000, 4792, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIXE
3, 0) = 0xEF715000
close(4) = 0
open("/usr/lib/libdl.so.1", O_RDONLY) = 4
fstat(4, 0xEFFFF064) = 0
mmap(0xEF7B0000, 4096, PROT_READ|PROT_EXEC, MAP_SHARED|MAP_FIXED,
4, 0) = 0xEF7B0000
close(4) = 0
mprotect(0x00010000, 5243, PROT_READ|PROT_WRITE|PROT_EXEC) =
0
open("/usr/platform/SUNW,SPARCstation-10/lib/libc_psr.so.1",
O_RDONLY) Err#2 ENOENT
mprotect(0x00010000, 5243, PROT_READ|PROT_EXEC) = 0
close(3) = 0
getrlimit(RLIMIT_NOFILE, 0xEFFFEEE8) = 0
brk(0x00021BB0) = 0
brk(0x00023BB0) = 0
access("/usr/lib/locale/de/LC_CTYPE/ctype", 4) = 0
open("/usr/lib/locale/de/LC_CTYPE/ctype", O_RDONLY) = 3
read(3, " \ 0 ( (".., 521) = 521
close(3) = 0
access("/usr/lib/locale/de/LC_NUMERIC/numeric", 4) = 0
open("/usr/lib/locale/de/LC_NUMERIC/numeric", O_RDONLY) = 3
read(3, " , ", 2) = 2
close(3) = 0
open("/usr/lib/locale/de/LC_TIME/time", O_RDONLY) = 3
close(3) = 0
access("/usr/lib/locale/de/LC_COLLATE/coll.so", 4) Err#2 ENOENT
access("/usr/lib/locale/de/LC_COLLATE/CollTable", 4) = 0
open("/usr/lib/locale/de/LC_MONETARY/monetary", O_RDONLY) =
3
close(3) = 0
stat("/usr/lib/locale/de/", 0xEFFFEAE0) = 0
open("/usr/dt/lib/nls/msg/de/SUNW_SPRO_SC_libpc.cat.cat", O_RDONLY)
```



```
Err#2 ENOENT
open("/usr/dt/lib/nls/msg/C/SUNW_SPRO_SC_libpc.cat.cat", O_RDONLY)
Err#2 ENOENT
open("../../lib/locale/de/LC_MESSAGES/SUNW_SPRO_SC_libpc.cat",
O_RDONLY) Err#2 ENOENT
open("/usr/lib/locale/de/LC_MESSAGES/SUNW_SPRO_SC_libpc.cat",
O_RDONLY) Err#2 ENOENT
sigaction(SIGQUIT, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGQUIT, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGILL, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGILL, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGTRAP, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGTRAP, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGEMT, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGEMT, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGFPE, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGFPE, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGBUS, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGBUS, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGSEGV, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGSEGV, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGSYS, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGSYS, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGXCPU, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGXCPU, 0xEFFFF1B8, 0x00000000) = 0
sigaction(SIGXFSZ, 0x00000000, 0xEFFFF238) = 0
sigaction(SIGXFSZ, 0xEFFFF1B8, 0x00000000) = 0
ioctl(1, TCGETA, 0xEFFFE8FC) = 0
ioctl(1, TCGETA, 0xEFFFE8FC) = 0
0,00000000000000e+00
write(1, " 0 , 0 0 0 0 0 0 0 0 0".., 22) = 22
1,00000000000000e-01
write(1, " 1 , 0 0 0 0 0 0 0 0 0".., 22) = 22
2,00000000000000e-01
write(1, " 2 , 0 0 0 0 0 0 0 0 0".., 22) = 22
3,00000000000000e-01
write(1, " 3 , 0 0 0 0 0 0 0 0 0".., 22) = 22
4,00000000000000e-01
write(1, " 4 , 0 0 0 0 0 0 0 0 0".., 22) = 22
5,00000000000000e-01
write(1, " 5 , 0 0 0 0 0 0 0 0 0".., 22) = 22
6,00000000000000e-01
write(1, " 6 , 0 0 0 0 0 0 0 0 0".., 22) = 22
```

```
7,00000000000000e-01
write(1, " 7 , 0 0 0 0 0 0 0 0 0".., 22) = 22
8,00000000000000e-01
write(1, " 8 , 0 0 0 0 0 0 0 0 0".., 22) = 22
9,00000000000000e-01
write(1, " 9 , 0 0 0 0 0 0 0 0 0".., 22) = 22
1,00000000000000e+00
write(1, " 1 , 0 0 0 0 0 0 0 0 0".., 22) = 22
1,10000000000000e+00
write(1, " 1 , 1 0 0 0 0 0 0 0 0".., 22) = 22
1,20000000000000e+00
write(1, " 1 , 2 0 0 0 0 0 0 0 0".., 22) = 22
1,30000000000000e+00
write(1, " 1 , 3 0 0 0 0 0 0 0 0".., 22) = 22
1,40000000000000e+00
write(1, " 1 , 4 0 0 0 0 0 0 0 0".., 22) = 22
1,50000000000000e+00
write(1, " 1 , 5 0 0 0 0 0 0 0 0".., 22) = 22
1,60000000000000e+00
write(1, " 1 , 6 0 0 0 0 0 0 0 0".., 22) = 22
1,70000000000000e+00
write(1, " 1 , 7 0 0 0 0 0 0 0 0".., 22) = 22
1,80000000000000e+00
write(1, " 1 , 8 0 0 0 0 0 0 0 0".., 22) = 22
1,90000000000000e+00
write(1, " 1 , 9 0 0 0 0 0 0 0 0".., 22) = 22
2,00000000000000e+00
write(1, " 2 , 0 0 0 0 0 0 0 0 0".., 22) = 22
lseek(0, 0, SEEK_CUR) = 18206
_exit(0)
```

`ldd -r hello` zeigt die benötigten (dynamisch gelinkten) Bibliotheken:

```
libpc.so.4 => /opt/SUNWspro/lib/libpc.so.4
libc.so.1 =>  /usr/lib/libc.so.1
libdl.so.1 => /usr/lib/libdl.so.1
```

`truss hello` im Falle einer statisch gelinkten Applikation (erzeugt mittels `pc -Bstatic ...`):


```
execve("./hello", 0xEFFFF3A8, 0xEFFFF3B4) argc = 2
open("/dev/zero", O_RDONLY) = 3
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE,
3, 0) = 0xEF7C0000
close(3) = 0
getrlimit(RLIMIT_NOFILE, 0xEFFFEEE0) = 0
brk(0x00046D90) = 0
brk(0x00048D90) = 0
access("/usr/lib/locale/de/LC_CTYPE/ctype", 4) = 0
open("/usr/lib/locale/de/LC_CTYPE/ctype", O_RDONLY) = 3
read(3, "\ 0 ( (".., 521) = 521
close(3) = 0
access("/usr/lib/locale/de/LC_NUMERIC/numeric", 4) = 0
open("/usr/lib/locale/de/LC_NUMERIC/numeric", O_RDONLY) = 3
read(3, " , ", 2) = 2
close(3) = 0
open("/usr/lib/locale/de/LC_TIME/time", O_RDONLY) = 3
close(3) = 0
access("/usr/lib/locale/C/LC_CTYPE/ctype", 4) = 0
open("/usr/lib/locale/C/LC_CTYPE/ctype", O_RDONLY) = 3
read(3, "\ 0 ( (".., 521) = 521
close(3) = 0
access("/usr/lib/locale/C/LC_NUMERIC/numeric", 4) = 0
open("/usr/lib/locale/C/LC_NUMERIC/numeric", O_RDONLY) = 3
read(3, " .\ 0", 2) = 2
close(3) = 0
open("/usr/dt/lib/nls/msg/de/SUNW_SPRO_SC_libpc.cat.cat", O_RDONLY)
Err#2 ENOENT
open("/usr/dt/lib/nls/msg/C/SUNW_SPRO_SC_libpc.cat.cat", O_RDONLY)
Err#2 ENOENT
open("../../lib/locale/de/LC_MESSAGES/SUNW_SPRO_SC_libpc.cat",
O_RDONLY) Err#2 ENOENT
open("/usr/lib/locale/de/LC_MESSAGES/SUNW_SPRO_SC_libpc.cat",
O_RDONLY) Err#2 ENOENT
sigaction(SIGQUIT, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGQUIT, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGILL, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGILL, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGTRAP, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGTRAP, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGEMT, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGEMT, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGFPE, 0x00000000, 0xEFFFF230) = 0
```



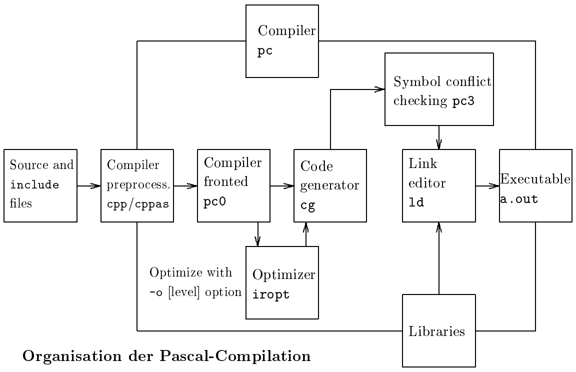
```
sigaction(SIGFPE, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGBUS, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGBUS, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGSEGV, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGSEGV, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGSYS, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGSYS, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGXCPU, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGXCPU, 0xEFFFF1B0, 0x00000000) = 0
sigaction(SIGXFSZ, 0x00000000, 0xEFFFF230) = 0
sigaction(SIGXFSZ, 0xEFFFF1B0, 0x00000000) = 0
ioctl(1, TCGETA, 0xEFFFE8FC) = 0
ioctl(1, TCGETA, 0xEFFFE8FC) = 0
0.00000000000000e+00
write(1, " 0 .  0 0 0 0 0 0 0 0 0".., 22) = 22
1.00000000000000e-01
write(1, " 1 .  0 0 0 0 0 0 0 0 0".., 22) = 22
2.00000000000000e-01
write(1, " 2 .  0 0 0 0 0 0 0 0 0".., 22) = 22
3.00000000000000e-01
write(1, " 3 .  0 0 0 0 0 0 0 0 0".., 22) = 22
4.00000000000000e-01
write(1, " 4 .  0 0 0 0 0 0 0 0 0".., 22) = 22
5.00000000000000e-01
write(1, " 5 .  0 0 0 0 0 0 0 0 0".., 22) = 22
6.00000000000000e-01
write(1, " 6 .  0 0 0 0 0 0 0 0 0".., 22) = 22
7.00000000000000e-01
write(1, " 7 .  0 0 0 0 0 0 0 0 0".., 22) = 22
8.00000000000000e-01
write(1, " 8 .  0 0 0 0 0 0 0 0 0".., 22) = 22
9.00000000000000e-01
write(1, " 9 .  0 0 0 0 0 0 0 0 0".., 22) = 22
1.00000000000000e+00
write(1, " 1 .  0 0 0 0 0 0 0 0 0".., 22) = 22
1.10000000000000e+00
write(1, " 1 .  1 0 0 0 0 0 0 0 0".., 22) = 22
1.20000000000000e+00
write(1, " 1 .  2 0 0 0 0 0 0 0 0".., 22) = 22
1.30000000000000e+00
write(1, " 1 .  3 0 0 0 0 0 0 0 0".., 22) = 22
1.40000000000000e+00
write(1, " 1 .  4 0 0 0 0 0 0 0 0".., 22) = 22
```



```
1.50000000000000e+00
write(1, " 1 .  5 0 0 0 0 0 0 0 0".., 22) = 22
1.60000000000000e+00
write(1, " 1 .  6 0 0 0 0 0 0 0 0".., 22) = 22
1.70000000000000e+00
write(1, " 1 .  7 0 0 0 0 0 0 0 0".., 22) = 22
1.80000000000000e+00
write(1, " 1 .  8 0 0 0 0 0 0 0 0".., 22) = 22
1.90000000000000e+00
write(1, " 1 .  9 0 0 0 0 0 0 0 0".., 22) = 22
2.00000000000000e+00
write(1, " 2 .  0 0 0 0 0 0 0 0 0".., 22) = 22
lseek(0, 0, SEEK_CUR) = 4840
_exit(0)
```

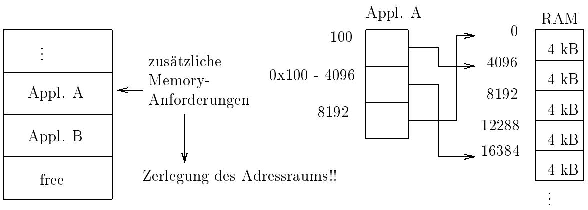
Für eine kurze Übersicht der durchgeführten System-Calls benutze man `truss -c hello`:

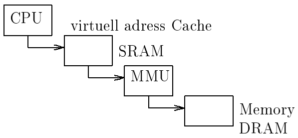
```
syscall      seconds  calls  errors

_exit            .00      1
read             .00      2
write            .00     21
open             .01     19      10
close            .01      9
brk              .00      2
stat             .00      1
lseek            .00      1
fstat            .00      4
access           .01      4       1
ioctl            .00      2
execve           .00      1
sigaction        .01     20
mmap             .01     12
mprotect         .00      2
munmap           .00      3
getrlimit        .00     17
--------------------------------------
sys totals:      .05   1057      11
usr time:        .04
elapsed:         .29
```



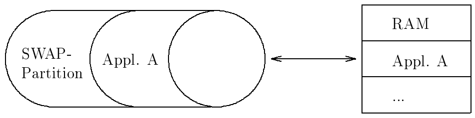
Organisation der Pascal-Compilation

Paging: Virtuelles Memory. *Problem:* Es ergibt sich bei wachsender Laufzeit auch ein erhöter Speicherbedarf!





Swapping: *Problem:* Der aktuelle Applikations-Hauptspeicherbedarf ist größer als das vorhandender RAM.



1. `.text` braucht nicht geswapt zu werden. Warum?
2. Mischung von Swapping und Paging:

   memory fault ⟶ geswappte Page ⟶ RAM

2.2 Hilfsmittel bei der Programmentwicklung

2.2.1 pxref – Querverweistabellen

```
%pxref x.p
```

2.2.2 pretty – automatisches „logisches“ Einrücken

`cat x.p|pretty|more`: automatisches Einrücken zur Überprüfung

```
program dynzeichenketten ( input, output ) ;

label 9999 ;

type Natural = 0 ..  maxint ;
Positive = 1 ..  maxint ;
Zeichenkette = ↑ ZKElement ;
ZKElement = record
      c :  char ;
      next :  Zeichenkette ;
end ;

var ZKette1 :  Zeichenkette :
```



```
procedure MakeZK ( Str :  array [ low ..  high ] :  integer ] of char ;
var ZK : Zeichenkette ) ;

var i :  integer ;
NeuesElement , LetztesElement :  Zeichenkette ;
begin
      new ( NeuesElement ) ;
      if ( NeuesElement = nil ) then begin
            writeln ('*** Fehler:  ungenügender Speicherplatz in MakeZK' ) ;
            goto 9999 ;
      end ;
      NeuesElement ↑ .  c := Str [ low ] ;
      NeuesElement ↑ .  next := nil ;
      ZK := NeuesElement ;
      LetztesElement := NeuesElement ;
      for i := succ ( low ) to high do begin
            new ( NeuesElement ) ;
            if ( NeuesElement = nil ) then begin
                  writeln ("*** Fehler:  ungenügender Speicherplatz in MakeZK' ) ;
                  goto 9999 ;
            end ;
            NeuesElement ↑ .  c := Str [ i ] ;
            NeuesElement ↑ .  next := nil ;
            LetztesElement ↑ .  next := NeuesElement ;
            LetztesElement := NeuesElement ;
      end ;
end ;


procedure SchreibZK ( ZK : Zeichenkette ) ;
begin ;
      while ( ZK <> nil ) do begin
            write ( ZK ↑ .  c ) ;
            ZK := ZK ↑ .  next ;
      end ;
end;


procedure LoescheZK (
var ZK : Zeichenkette :  Stelle :  Positive ; AnzZeichen :  Natural ) ;

var i, j :  Natural ;
Aktueller , Hilf , Naechster :  Zeichenkette ;
```



```
begin
      if ( Stelle = 1 ) then begin
            i := 0 ;
            while ( ZK <> nil ) and ( i < AnzZeichen ) do begin
                  Naechster := ZK ↑ .  next ;
            ⋮
```

2.2.3 vgrind

1. Index aller Dateien eines Projekts
   `%ctags -v *.p *.inc |sort -f > index`
   `%vgrind -lp -x index | /usr/lib/lp/postscript/dpost > index.ps`
2. `vgrind -lp scalarprod.p | /usr/lib/lp/postscript/dpost > scalarprod.ps`

funcpar.p

{ *Testprogramm fuer konforme Funktionen als formalen Parameter* }
{ *File: funcpar.p* *Autor: HJB* *Datum: 10/06/94* }
{ }
program FuncParameter(input, output);
const
  low1 = 0 ; high1 = 1;
type
  Vektor2dim = **array**[low1..high1] **of** real ;
var
  Index2dim : low1..high1 ;
  A2dim : Vektor2dim ;

{ }
{ *Klasse von durch Funktionen parametrisierten Methoden* }
{ }
function Mittelwert (a : **array**[1 .. h : integer] **of** real;
    function KomponentenOperation(x : real) : real ;
    function NormierungsOperation(x : real) : real
) : real ;
var
  Index : integer ;
  Akku : real ;
begin
  Akku := 0.0;
  for Index :=1 **to** h **do**

```
            Akku := Akku + KomponentenOperator(a[Index]) ;
        Mittelwert := NormierungsOperator(Akku) ;
end: { function ScProd }

{                                                              }
{        Funktionen fuer die Benutzung als aktueller Parameter. }
{                                                              }
{ a) Komponenten-Operationen:                                  }
{                                                              }
function Quadrat( x : real ) : real :
begin
        Quadrat := x*x ;
end;
{                                                              }
{ b) Normierungs-Operationen:                                  }
{                                                              }
function QuadrWurzel( x : real ) : real ;
begin
        QuadrWurzel := sqrt(x) ;
end;
{                                                              }

begin

        A2dim[0] := 3.0 ;
        A2dim[1] := 4.0 ;

        writeln('Ergebnis fuer 2-dim. Testvektoren: ',
                Mittelwert(A2dim, Quadrat, QuadrWurzel));

end. { program FuncParameter }
```

2.2.4 Zeitmessung

```
%time xyz
%/usr/bin/time xyz
%truss -c xyz
```

In Pascal z.B.:

```
⋮
s := Clock ;    {user time in millisec}
⋮
```


oder

```
⋮
s := sysclock ;    {system time in millisec}
⋮
```

2.2.5 Profiling

1. `% pc -g xxx.p`
   `sparworks&`

   debugger — starten

   Execute

   Collector

   (debugger) debug a.out

   (debugger) a.out

   analyser — starten

   load — anklicken

2. prof
   ```
   % pc -p xxx.p
   % a.out
   % prof a.out
   ```
3. gprof
   ```
   % pc -gp xxx.p
   % a.out
   % gprof a.out
   ```
4. tcav
   ```
   % pc -a -o xxx xxx.p
   % xxx
   % tcov xxx.p
   % more xxx.tcov
   ```


Für C/C++ gibt es noch die Option `lprof`.

2.2.6 Browsing

```
% pc -sb xxx.p
% sbrowser &
```

2.3 Übersicht Systemsoftware

Systemsoftware

Betriebsystem

Programmentwicklung:
- Editor, Shells
- Compiler
- Interpreter
- Linker
- Bibliotheksverwaltung
- Debugger
- Testwerkzeuge
- Design (CASE-) Tools
- GUI-Generatoren
- Dokumentationshilfen
- Spezifikations-/ Verifikationstools

Datenbanken

2.4 Betriebssystem-Kernel

Hardware (Device-) Treiber
Dateisubsystem
- Hierarchistisches Filesystem: Inhaltsverzeichnisse, Dateien
- Geräte *special files*: charcter-Devices, block-Devices
- pipes, sockets / streams

Prozeßsubsystem
- Memorymanegement (virtuelles Memory)
- Scheduler (Multiprozessing, Multithreading)
- IPC

Graphiksubsysteme
- X-Windows, Motif, OpenWindows
- XGL, XIL
- PHIGS, PEX
- KCMS (Kodate Color Management System)
- Display Postscript

Datenbankdienste