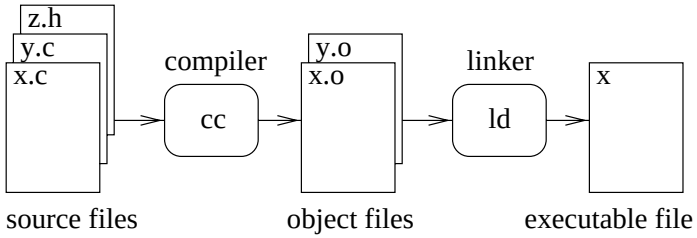
Lecture overview

- Programmers interface
- System calls
- C interface library
- Examples

Using C language



compilation – `cc -c x.c`
linking – `cc x.o y.o -o x`

Simple C program

```
#include <stdio.h>

int main( int narg, char *argv[] )
{
printf( "Number of arguments = %d\n", narg );
for (i=0; i<narg; i++)
  {
  printf( "Argument [%d] = %s\n" argv[i] );
  }
}
```


Compilation and results of simple C program

```
csd<jurek>(56)$ cc x.c -o x

csd<jurek>(57)$ ./x a b c
Number of arguments = 4
Argument [0] = ./x
Argument [1] = a
Argument [2] = b
Argument [3] = c
```

System calls 1

- `int chdir(const char *path)`
  Change working directory.
  Example:
  `chdir( "../bin" );`
- `int chown((const char *path, uid_t owner, gid_t group);`
  Change file ownership.
  Example:
  `chown( "file.c", 302, 50 );`

System calls 2

- `int chmod(const char *path, mode_t mode)`
  Change file permissions.
  Examples:
  `chmod( "file.c", S_IRUSR );`
  result: `-r-------- 1 jurek staff 8 Nov 15 10:17 file.c`
  `chmod( "../x.c", S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH );`
  result: `-rw-r--r-- 1 jurek staff 8 Nov 15 10:17 x.c`
  `chmod( "abc", 0754 );`
  result: `-rwxr-xr-- 1 jurek staff 8 Nov 15 10:17 abc`

System calls 3

File modes:
S_ISUID 04000 Set user ID on execution.
S_ISGID 02000 Set group ID on execution
S_IRWXU 00700 Read, write, execute by owner.
S_IRUSR 00400 Read by owner.
S_IWUSR 00200 Write by owner.
S_IXUSR 00100 Execute (search if a directory) by owner.
S_IRWXG 00070 Read, write, execute by group.
S_IRGRP 00040 Read by group.
S_IWGRP 00020 Write by group.
S_IXGRP 00010 Execute by group.
S_IRWXO 00007 Read, write, execute (search) by others.
S_IROTH 00004 Read by others.
S_IWOTH 00002 Write by others.
S_IXOTH 00001 Execute by others.

System calls 4

- `int link(const char *existing, const char *new);`
  Create hard link to existing file.
  Example:
  `link( "file.c", "link-to-file.c" );`
- `int symlink(const char *name1, const char *name2);`
  Create symbolic link to a file or directory.
  Example:
  `symlink( "file.c", "symlink.c" );`

System calls 5

- `int mkdir(const char *path, mode_t mode);`
  Create new directory.
  Example:
  ```
  mkdir( "subdir", S_IRWXU | S_IRGRP | S_IXGRP );
  ```
  result:
  ```
  drwxr-x---  1 jurek  staff  8 Nov 15 10:17 subdir
  ```
- `int rmdir(const char *path);`
  Remove a directory.
  Example:
  ```
  rmdir( "subdir" );
  ```


System calls 6

- `int rename(const char *old, const char *new);`
  Rename a file or directory.
  Example:
  ```
  rename( "file.c", "big-file.c" );
  ```
- `int unlink(const char *path);`
  Remove directory entry.
  Example:
  ```
  unlink( "file.c" );
  ```


System calls 7

- `int stat(const char *path, struct stat *buf);`
  Get information about the file.
  Example:

```
#include <sys/types.h>
#include <sys/stat.h>

struct stat buffer;

stat( "file.c", &buffer );
printf( "File size = %ld bytes\n", buffer.st_size );
```


System calls 8

- `time_t time(time_t *tloc);`
  Get current time (in seconds since 0:0:0 UTC, January 1, 1970.
- `char *ctime(const time_t *clock);`
  Convert time to string.
  Example:
  ```
  printf( "Today is %s", ctime( time(NULL) ) );
  ```
  result: `Today is  Fri Sep 13 00:00:00 1994`
- `struct tm *localtime(const time_t *clock);`
  Convert time to second, minute, hour, day, moth and year.
  Example:
  ```
  #include <time.h>
  struct tm *t;
  t = localtime( time( NULL ) );
  printf( "Today is 19%02d.%02d.%02d\n",
  t->tm_year, t->tm_mon, t->tm_mday );
  ```