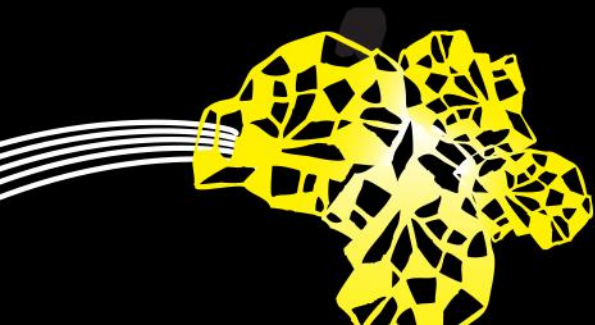
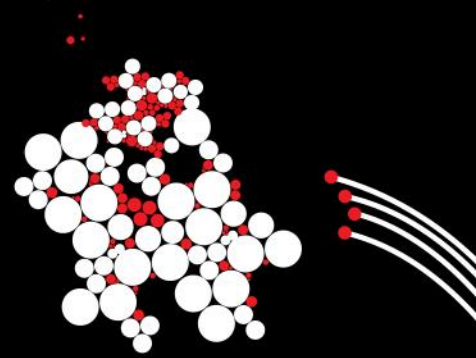


UNIVERSITY OF TWENTE.

## OPERATING SYSTEMS

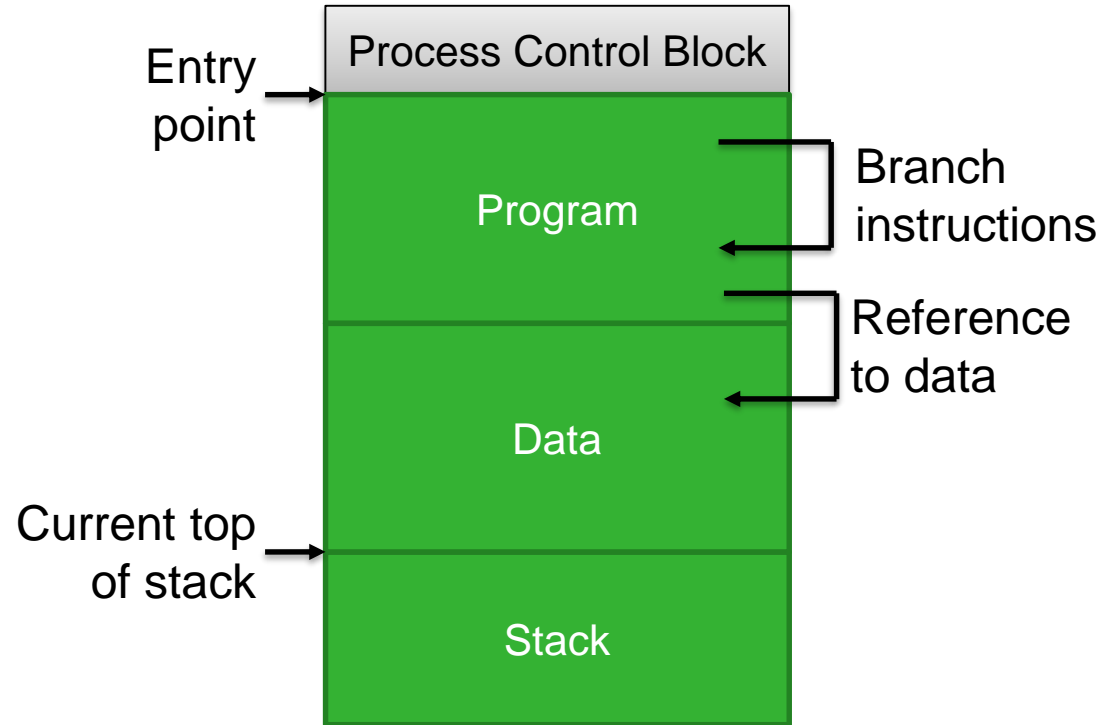
### MEMORY

ERIK TEWS <[e.tews@utwente.nl](mailto:e.tews@utwente.nl)>



# MEMORY MANAGEMENT REQUIREMENTS

- Logical organization
- Physical organization
- Relocation
- Protection
- Sharing
- OS+HW cooperation
- Store OS itself

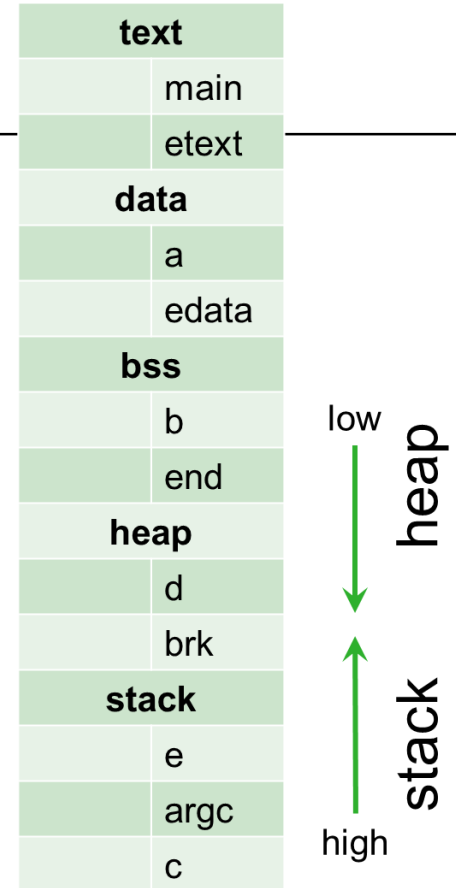


# ADDRESS SPACE MANAGEMENT

- Output?
- gcc AddressSpace.c
- ./a.out

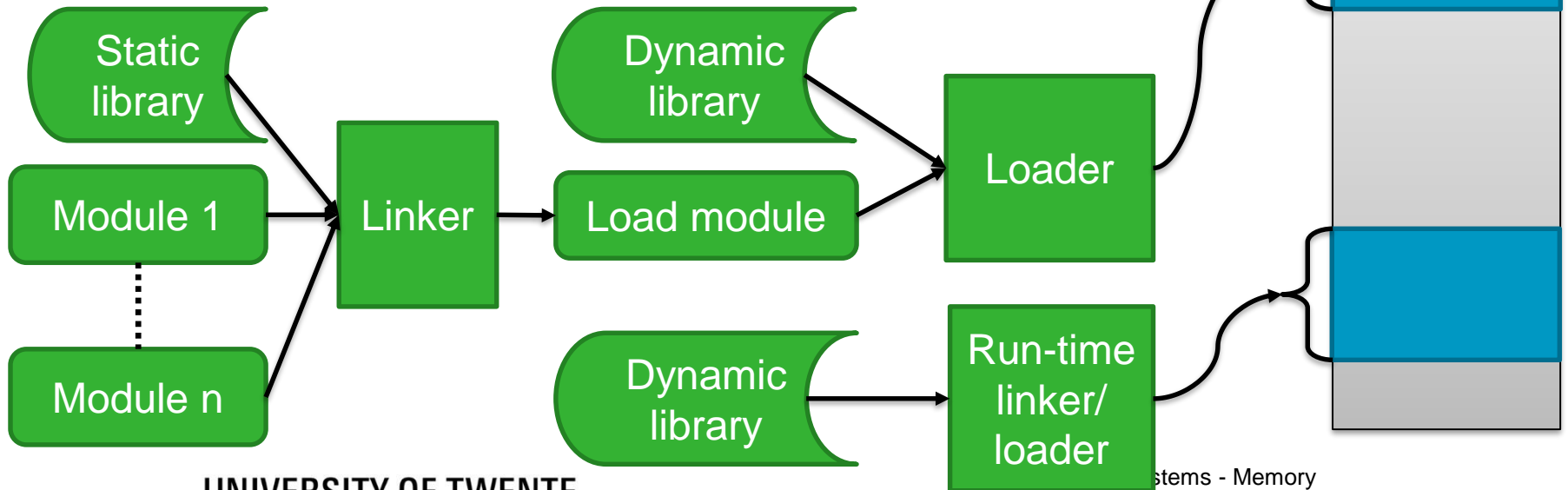
```
extern char etext, edata, end;
int a = 0xaaaa, b;
int main(int argc, char * argv[]) {
    int c = 0xcccc;
    int *d_ptr = (int*) malloc(sizeof(int));
    int *e_ptr = (int*) alloca(sizeof(int));
    b = 0xbbbb;
    *d_ptr = 0xdddd;
    *e_ptr = 0xeeee;
    printf("%p:a=%0x\n", &a, a); ...
}
```

UNIVERSITY OF TWENTE.



# LINKING AND LOADING OF USER PROCESSES

- Resolving symbolic references (why?)
- Collecting the modules at different times (why?)



# MEMORY PARTITIONING TECHNIQUES

---

More sophisticated

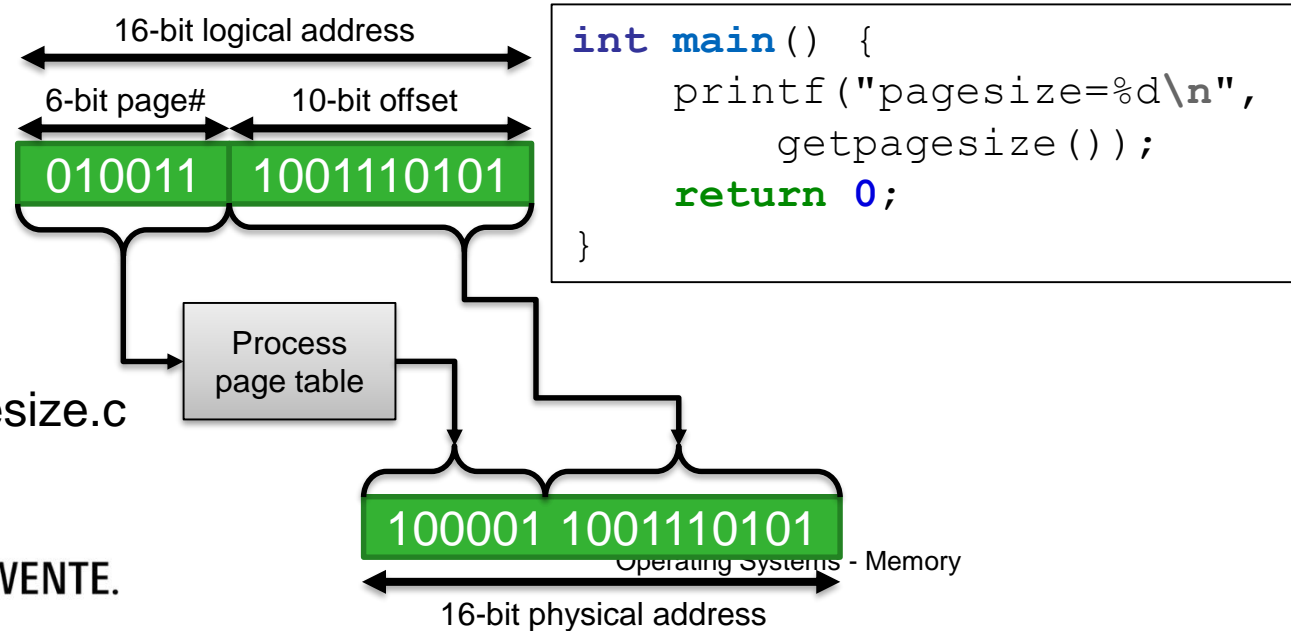


- Process loaded up front
  - Static/dynamic partitioning: whole process in whole partition
  - Segmentation: process divided into segments and memory divided into dynamic partitions
- Process loaded incrementally
  - Paging: process and memory divided in equal size pages
- Algorithms for
  - Placement and or compaction to reduce fragmentation
  - Swapping when the memory is getting full

# LOGICAL ADDRESSES VS PHYSICAL ADDRESSES

- Partitioning: physical address = relative address + base register
- Paging: page table per process mapping logical pages to physical frames

- Sharing?
- Protection?



- gcc Getpagesize.c
- ./a.out

# ProcessLayout.c

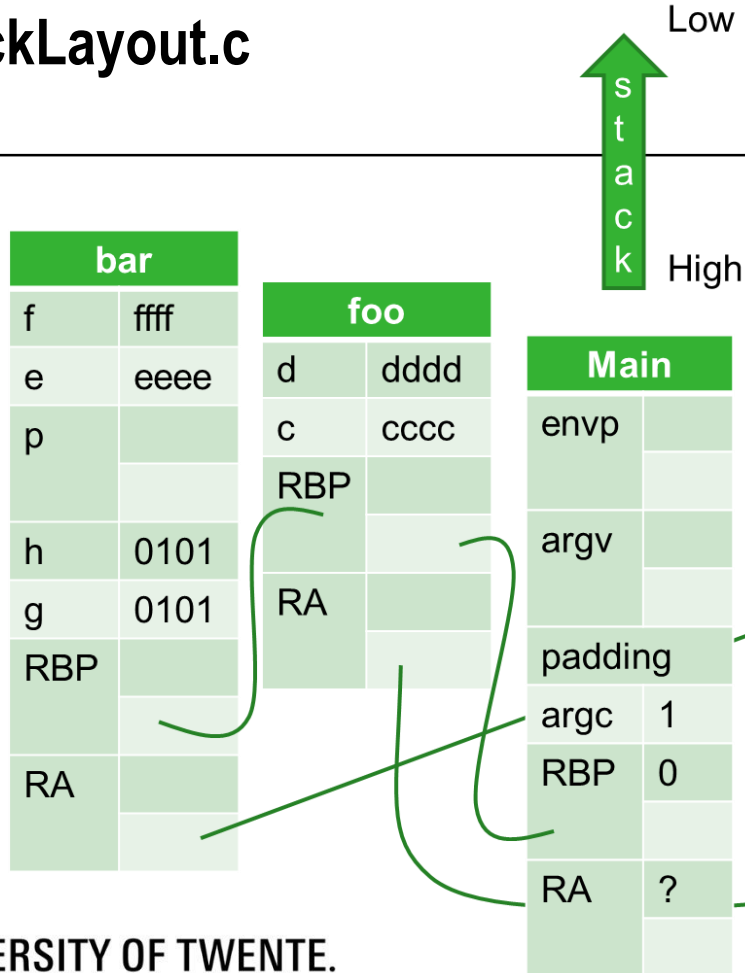
- gcc ProcessLayout.c -lpthread
- ./a.out
- [goo.gl/9JY4Os](https://goo.gl/9JY4Os)



```
void *tproc(void *ptr) {
    printf("tid=%p\n", pthread_self());
    fflush(stdout);
    sleep(1);
}

int main(int argc, char * argv[]) {
    int i ;
    pthread_t thread;
    printf("pid=%d, tid=%p\n",
        getpid(), pthread_self());
    fflush(stdout);
    for(i = 0; i < N; i++) {
        void *p = sbrk(M) ;
        pthread_create(&thread, NULL,
            &tproc, NULL);
        printf("sbrk %p\n", p);
        sleep(1);
    }
    return 0;
}
```

# StackLayout.c



```
#define N 4
int *q;

void bar(int e, int f) {
    int g = 0x1010, h = 0x0101;
    int *p;
    for(p=&f-N;p<=q;p++){
        printf("%p\t%0x\n", p,
            *p);
    }
}

void foo(int c, int d) {
    bar(0xeeee, 0xffff);
}

int main(int argc, char * argv[],
        char *envp[]) {
    int a = 0xaaaa, b = 0xbbbb;
    q = &a+N;
    foo(0xcccc, 0xdddd);
    return 0;
}
```

# SUMMARY

---

- Memory is a resource that can be partitioned
- Requires hardware support
- As many processes in the memory as possible
- Paging divides a process image in fixed size pages, segmentation in variable sized segments. Can be combined
- Linux uses paging