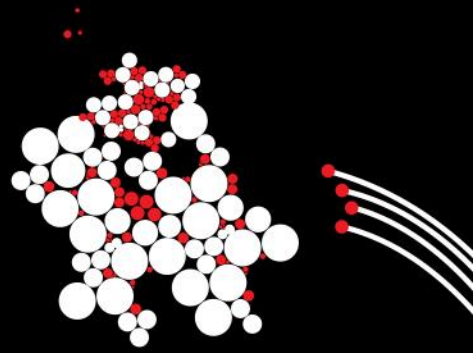
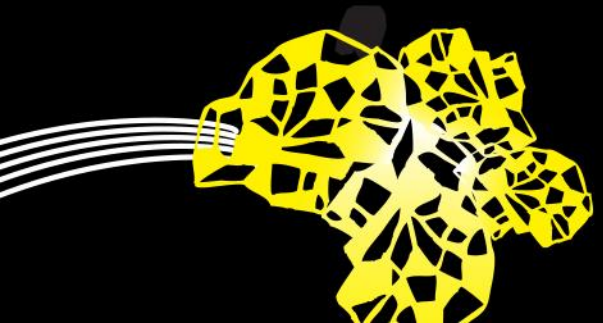
UNIVERSITY OF TWENTE.



OPERATING SYSTEMS

VIRTUAL MEMORY

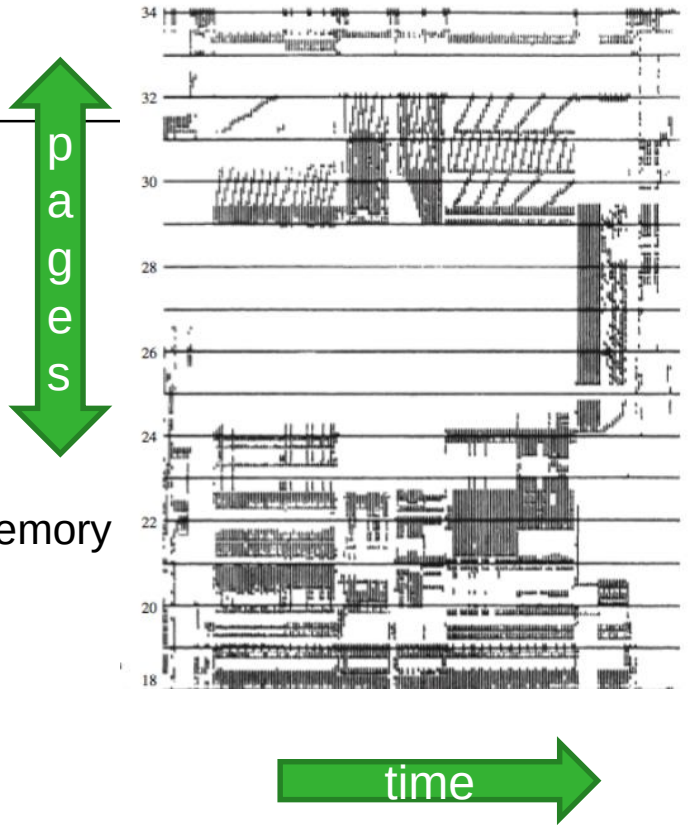
ERIK TEWS <e.tews@utwente.nl>



PRINCIPLE OF LOCALITY

MEMORY REFERENCES TEND TO CLUSTER

- Only resident set of pages in memory
  - Page fault brings in a missing page
  - Page fault may need to free a frame
- Advantages
  - Process size can be > memory size
  - The resident set of many processes in memory
- Challenges
  - OS has to predict future
  - Avoid thrashing
  - Concurrency may spoil locality



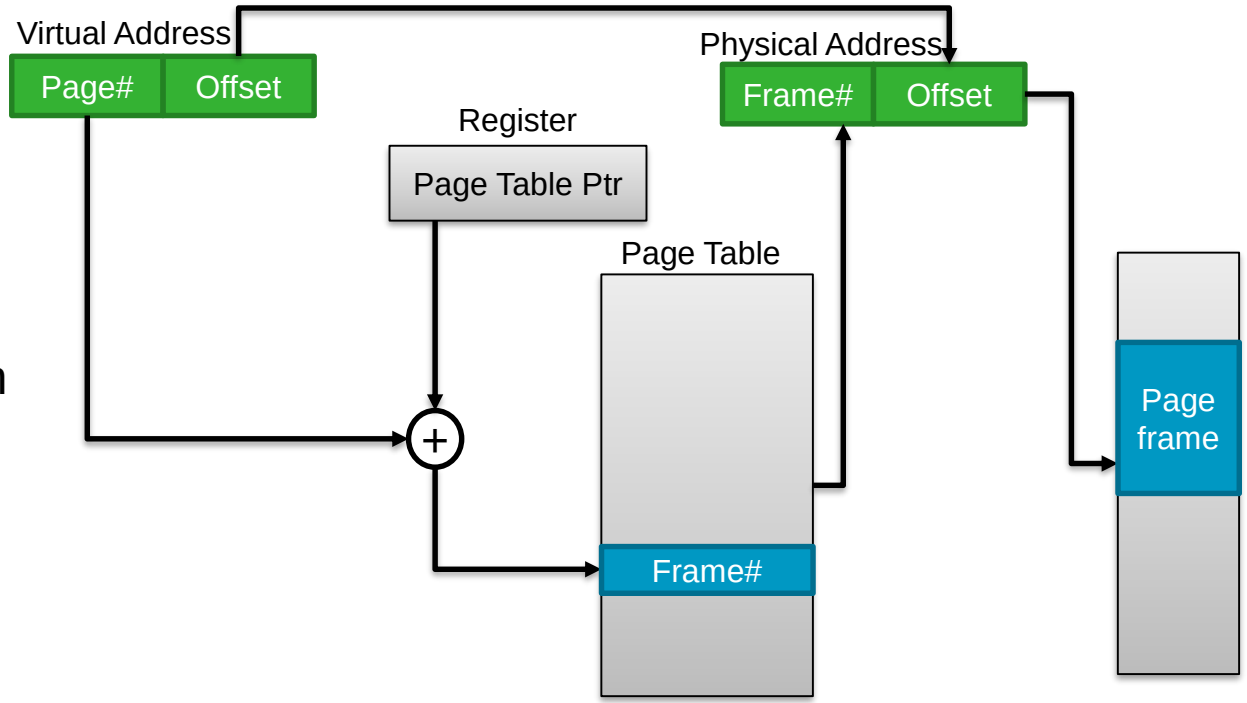
PRINCIPLE OF SEPARATION

- A policy describes what is allowed to be done
- A mechanism describes how it is done

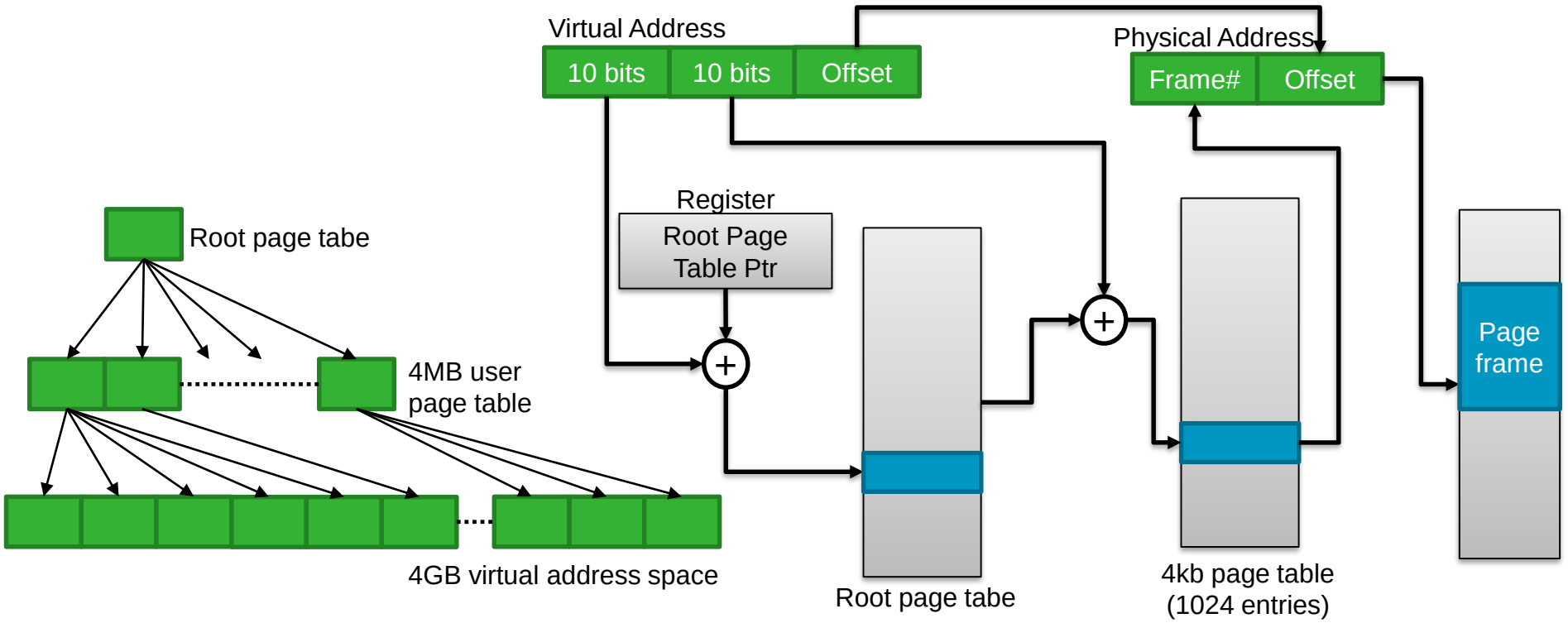
```
pi@mighty:~ $ ls -l
total 172
drwxr-xr-x 3 pi pi  4096 Oct 12 08:24 Download
-rw-r--r-- 1 pi pi 40884 Oct  1 20:23 Download-v1.zip
-rw-r--r-- 1 pi pi 44958 Oct  8 21:50 Download-v3.zip
-rwxr-xr-x 1 pi pi  8220 Oct  4 18:45 ForkVsThread
```

PAGING ADDRESS TRANSLATION MECHANISM

- Page table per process (how large?)
- Cache page table entries in Translation Look aside buffer (TLB)



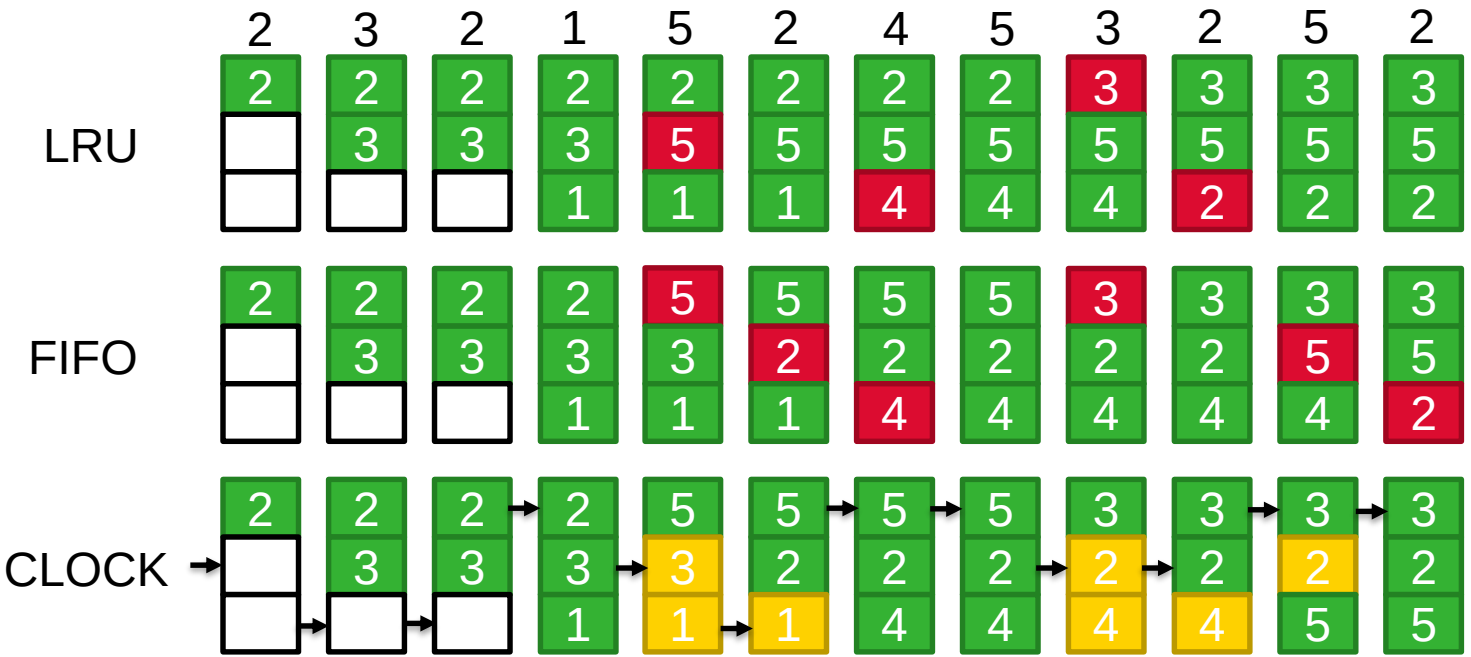
HIERARCHICAL PAGE TABLE



POLICIES

- Replacement policies
- Optimal, LRU, FIFO, and Clock (more…)
- Resident set policies
- Fixed and variable (more…)
- Others…

REPLACEMENT STRATEGIES EXAMPLE



RESIDENT SET MANAGEMENT POLICIES

- Choices: Fixed vs variable allocation and Local vs global scope
- Working set size depends on the time t (can we measure this?)
- Use page fault frequency changes as a trigger (how?)

PAGE FAULTS

- gcc Getrusage.c
- ./a.out
- ./a.out xx

- cd /usr/include/linux/
- more resource.h
- man mlock

```
int main(int argc, char* argv[]) {
    int cnt, flt;
    char buffer[N*P];
    struct rusage usage;
    if(argc > 1) mlock(buffer, N*P);
    getrusage(RUSAGE_SELF, &usage);
    flt = usage.ru_minflt;
    for (cnt=0; cnt < N*P; cnt++) {
        buffer[cnt] = 0;
        getrusage(RUSAGE_SELF, &usage);
        if( flt != usage.ru_minflt ) {
            flt = usage.ru_minflt;
            /* save cnt and flt to print later */
        }
    }
    /* print table of cnt and flt */
}
```

SHARING

- Output?
- gcc Mmap.c
- strace ./a.out Mmap.c Foo.c
- ./a.out Mmap.c Bar.c xx
- diff Mmap.c Foo.c
- diff Mmap.c Bar.c
- od –c Bar.c
- man mmap

```
int main(int argc, char *argv[]) {
 void *src, *tgt;
    int in = open(argv[1], O_RDONLY);
    int out = open(argv[2], O_RDWR|
        O_CREAT|O_TRUNC, 0666);
    int flags = (argc > 3 ?
        MAP_PRIVATE :  MAP_SHARED);
    size_t sz = lseek(in, 0, SEEK_END);
    lseek(out, sz - 1, SEEK_SET);
    write(out, "\0", 1);
    src = mmap(NULL, sz, PROT_READ,
        MAP_PRIVATE, in, 0);
    tgt = mmap(NULL, sz, PROT_WRITE,
        flags, out, 0);
    memcpy(tgt, src, sz);
    munmap(src, sz);
    munmap(tgt, sz);
    close(in);
    close(out);
    return 0;
}
```

SUMMARY

- Principles of locality and separation
- Memory management necessary to match slow I/O to fast processor
- Clean separation of logical from physical addresses
- Every modern system has virtual memory