

NOTICE – CONT'D

- Deadlines
 - (4/13 11:59 PM) Programming assignment 1
 - (4/20 11:59 PM) Midterm quiz 1

RECAP – CONT'D

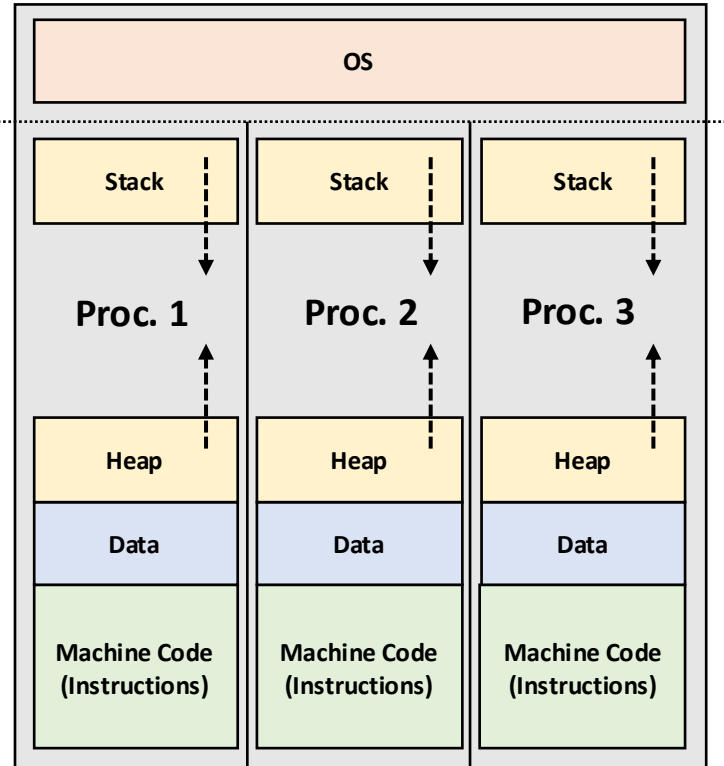
- Part I: Threads
 - Provide abstraction
 - What is a thread?
 - How is it different from a process?
 - How does OS run threads?
 - Offer standard libraries
 - How do we create/run/kill a thread?
 - How does OS manage the thread(s) we ran?
 - Manage resources
 - (Note) We will talk about this in the “scheduling” and “synchronization” classes

PROVIDE ABSTRACTION: A THREAD

- Thread
 - **Definition:** a smallest schedulable execution context
 - **Terminology:**
 - Smallest: it's much light-weight than a process
 - Schedulable execution context: one thread can run on a CPU at a time

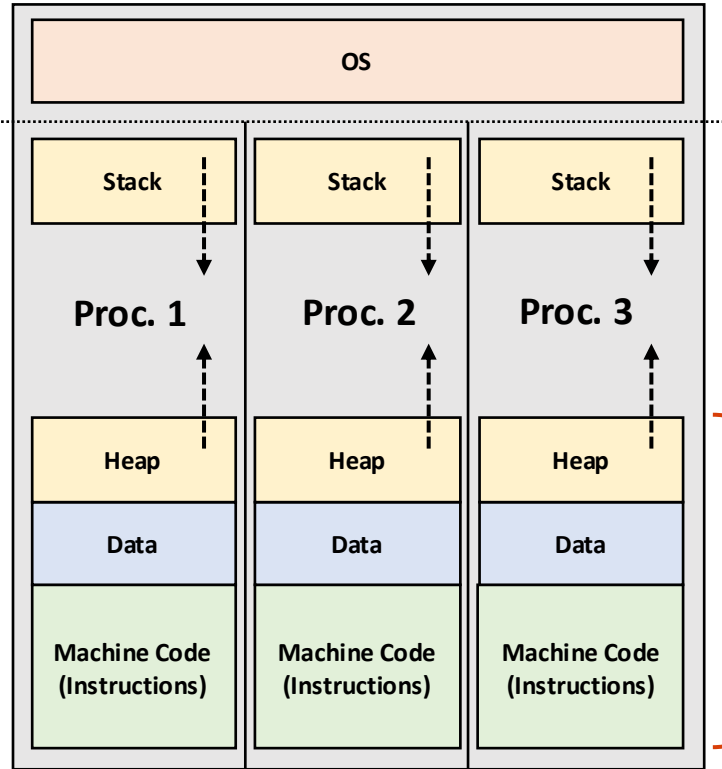
PROVIDE ABSTRACTION: A THREAD – CONT'D

Processes on memory

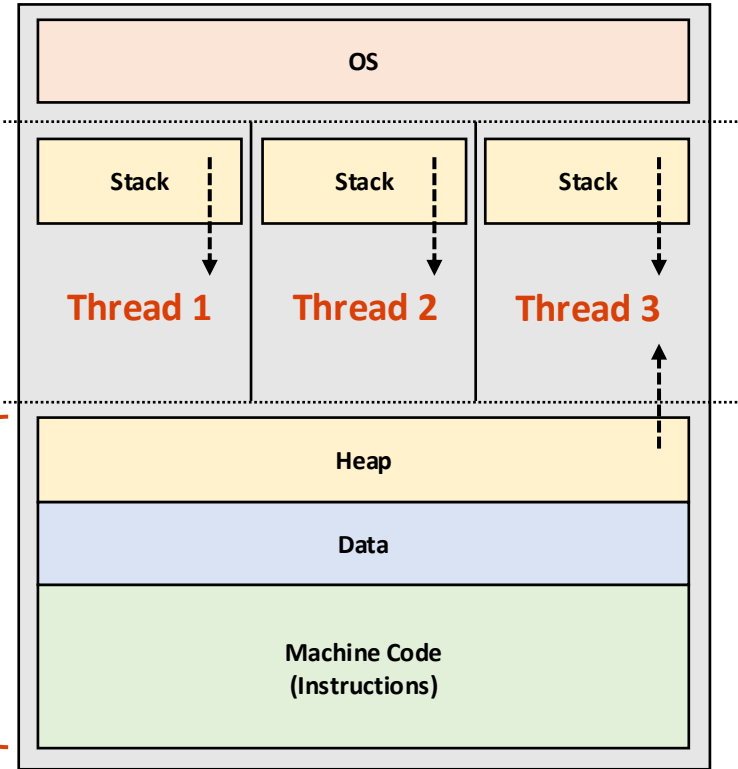


PROVIDE ABSTRACTION: A THREAD – CONT'D

Processes on memory



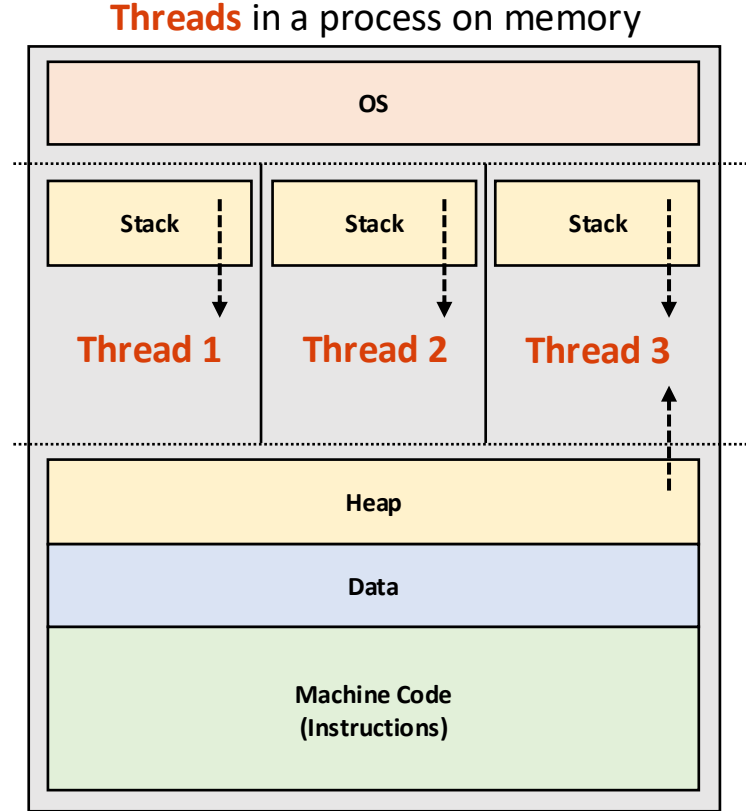
Threads in a process on memory



Reduce Duplications

PROVIDE ABSTRACTION: HOW IS IT DIFFERENT FROM A PROCESS?

- Threads share:
 - **Code** and **data** segments
 - **Heap** memory (ex. global variables)
 - Open files (ex. I/O access points)
- Threads **do not** share:
 - **Stack** segments, e.g.:
 - arguments passed when we launch them
 - local variables we initialize within them
 - return address, when they terminate (**OS II**)
 - Running contexts, e.g.:
 - process state
 - stack pointer
 - ...



TOPICS FOR TODAY

- Part I: Threads
 - Provide abstraction
 - What is a thread?
 - How is it different from a process?
 - How does OS run threads?
 - Offer standard libraries
 - How do we create/run/kill a thread?
 - How does OS manage the thread(s) we ran?
 - Manage resources
 - (Note) We will talk about this in the “scheduling” and “synchronization” classes

OFFER STANDARD INTERFACE

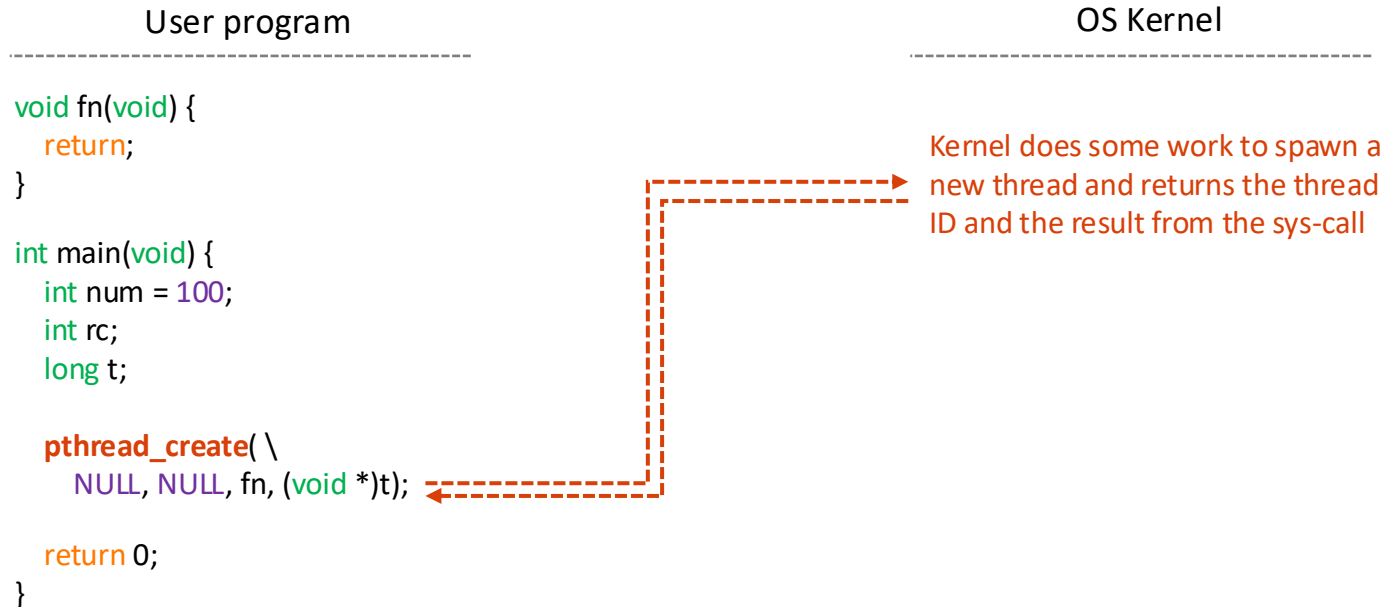
- How do we run a thread?
 - **System calls**
 - OS provide a set of system calls to control thread execution

OFFER STANDARD INTERFACE: **THREAD-SPECIFIC** SYSTEM CALLS

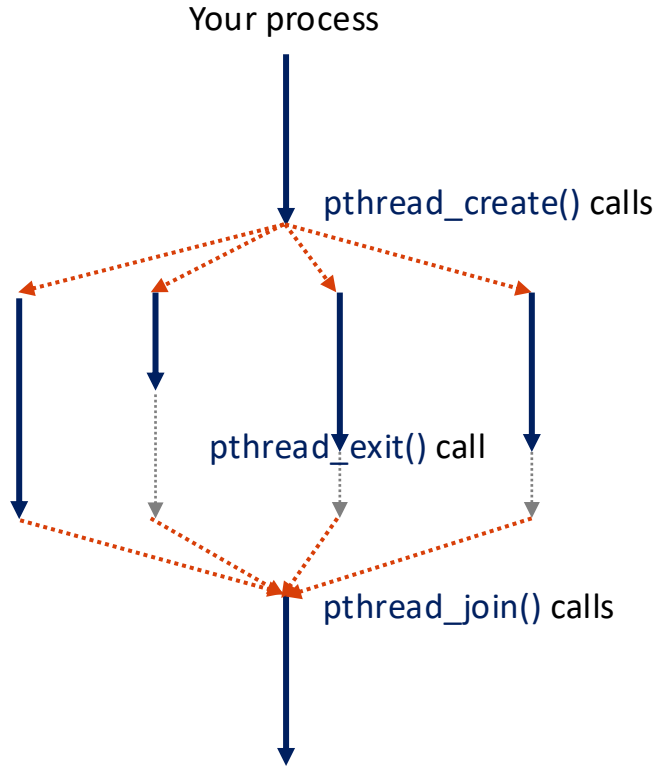
- Thread-specific system calls
 - **pthread_create**(thread, attribute, subroutine, subroutine-arguments);
 - Create a new thread executing the *subroutine* in the current process
 - Returns zero if it's successful, otherwise it returns [errno](#)
 - **pthread_exit**(return-value);
 - Terminate the thread and returns the *return-value* to any successful join
 - Note: If a thread terminates, it will be automatically called and always return success
 - **pthread_join**(thread, return-value-loc);
 - Suspend execution of the calling thread until the *thread* terminates
 - Once the thread terminates, the function will copy the return value to *return-value-loc*
 - Returns zero if it's successful, otherwise it returns an error

OFFER STANDARD INTERFACE: THREAD-SPECIFIC SYSTEM CALLS

- System call
 - **Example:** `pthread_create(...)`



THREAD PROGRAMMING: FORK-JOIN PATTERN



- **Fork - Join** Pattern
 - **Fork**: Main process creates a set of sub-(or child)-threads that runs a function
 - Each thread exits if the function returns
 - **Join**: Main waits until all the threads exit
- **Example**: download a large file
 - Splits a file into smaller chunks
 - Create a thread for downloading each
 - Sum-up all the downloaded chunks and combine them to create a single large file

OFFER STANDARD INTERFACE: THREAD-SPECIFIC SYSTEM CALLS

- Thread sample code in C
 - How many threads are there?
 - Which thread is created first?
 - Which thread is created last?
 - Which thread runs first/last?
 - What'd be an order of thread joins?
 - What will happen if we run this again?

```
static int value = 128;

void *subroutine (void *threadid) {
    long tid = (long) threadid;
    printf("Thread ID [%lx], value [%d]\n", tid, value ++);
}

int main(int argc, char *argv[]) {
    long t;
    int nthreads = 3;

    pthread_t *threads = (pthread_t *) malloc(nthreads * sizeof(pthread_t));
    memset(threads, 0x00, nthreads * sizeof(pthread_t));

    for (t = 0; t < nthreads; t++) {
        int rc = pthread_create(&threads[t], NULL, subroutine, (void *)t);
        if (rc) {
            printf("[Error] return code is: %d, abort.\n", rc);
            exit(-1);
        }
    }

    for (t = 0; t < nthreads; t++)
        pthread_join(threads[t], NULL);

    return 0;
}
```

OFFER STANDARD INTERFACE: THREAD-SPECIFIC SYSTEM CALLS

- Thread sample code in C
 - How many threads are there?
 - Which thread is created first?
 - Which thread is created last?
 - Which thread runs first/last?
 - What'd be an order of thread joins?
 - What will happen if we run this again?

Possible execution result:

Thread ID [0], value [128]

Thread ID [2], value [129]

Thread ID [1], value [130]

```
static int value = 128;

void *subroutine (void *threadid) {
    long tid = (long) threadid;
    printf("Thread ID [%lx], value [%d]\n", tid, value++);
}

int main(int argc, char *argv[]) {
    long t;
    int nthreads = 3;

    pthread_t *threads = (pthread_t *) malloc(nthreads * sizeof(pthread_t));
    memset(threads, 0x00, nthreads * sizeof(pthread_t));

    for (t = 0; t < nthreads; t++) {
        int rc = pthread_create(&threads[t], NULL, subroutine, (void *)t);
        if (rc) {
            printf("[Error] return code is: %d, abort.\n", rc);
            exit(-1);
        }
    }

    for (t = 0; t < nthreads; t++)
        pthread_join(threads[t], NULL);

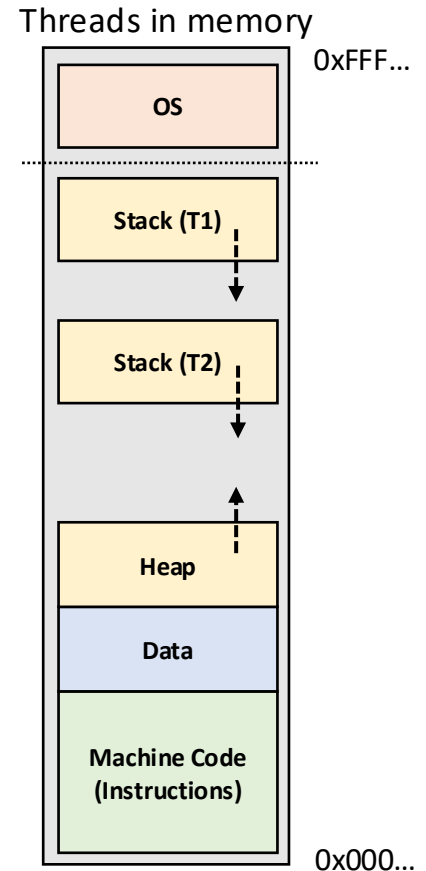
    return 0;
}
```

OFFER STANDARD INTERFACE: HOW OS MANAGES THREADS?

- **(Linux) OS**
 - A thread is treated as the same as a process
 - (Linux) thread control block \approx process context
- A thread can have **three states**:
 - **Ready**: a thread is created and ready to run, but not running now
 - **Running**: a thread running now
 - **Blocked**: a thread is unable to run (terminated or errors)

OFFER STANDARD INTERFACE: HOW OS MANAGES THREADS?

- Mem layout with two threads
 - Each thread has its own stack
 - Data, code and heap are shared between the two



CS 374: OPERATING SYSTEMS I

PART I – CONTEXT SWITCHING

M/W 12:00 – 1:50 PM (LINC #200)

Sanghyun Hong

sanghyun.hong@oregonstate.edu



Oregon State
University

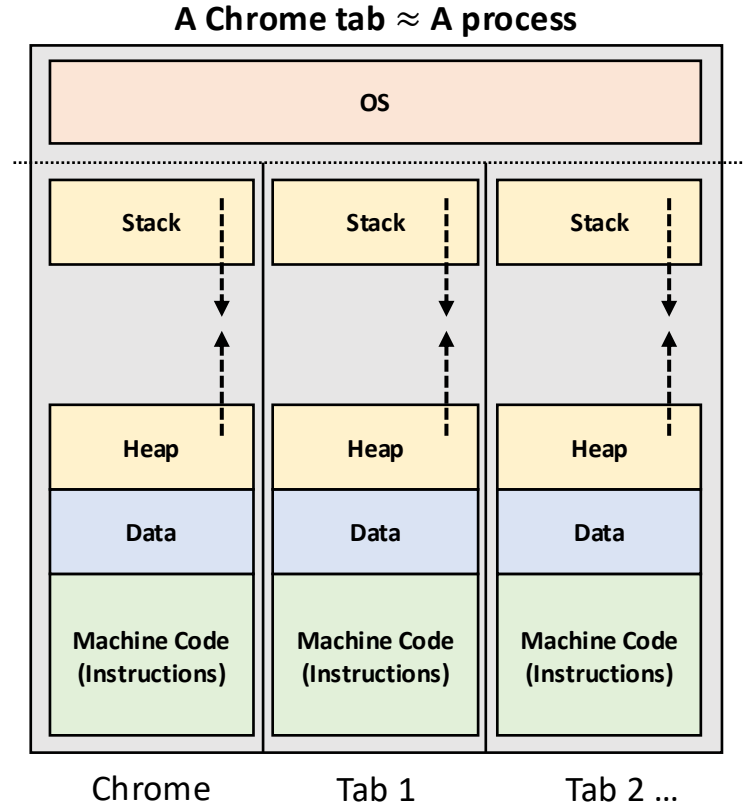


TRUE AI
Trustworthy and Responsible AI

PROBLEM: MULTIPLE PROGRAM, YET LIMITED PROCESSORS

- Your Chrome browser:
 - Open multiple websites (tabs)
 - Tab 1: Open Canvas website
 - Tab 2: Stack Overflow
 - Tab 3: Discord website
 - ... (many more 10+)
 - 4-8 CPUs (Processors)

How Can OS Address This Problem?



PROBLEM: MULTIPLE PROGRAM, YET LIMITED PROCESSORS – CONT'D

- Solution: (process) scheduling
 - **Goal:**
 - generate illusion that you're running 100+ processes at the same time
 - But in truth, it's not

SCHEDULING EXAMPLE: HIGH-LEVEL VIEW

- 3 Processes in Chrome:
 - **P1**: Download movies
 - **P2**: Open Canvas
 - **P3**: Search StackOverflow

- Example

- **New** :
- **Ready**:
- **Run** :
- **Wait** :
- **Term..**:

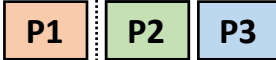


- **Scenario**: open a website for downloading movies

SCHEDULING EXAMPLE: HIGH-LEVEL VIEW

- 3 Processes in Chrome:
 - **P1**: Download movies
 - **P2**: Open Canvas
 - **P3**: Search StackOverflow

- Example

- **New** : 
- **Ready**:
- **Run** :
- **Wait** :
- **Term..**:

_____→ **Time**

- **Scenario**: the website opened and open two other websites

SCHEDULING EXAMPLE: HIGH-LEVEL VIEW

- 3 Processes in Chrome:
 - **P1**: Download movies
 - **P2**: Open Canvas
 - **P3**: Search StackOverflow

- Example

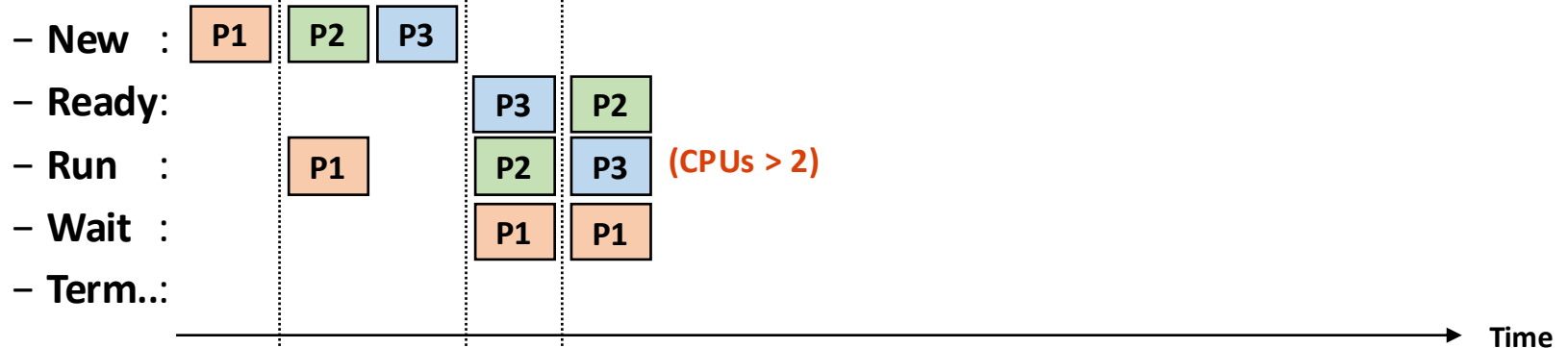


- **Scenario**: downloads started, and you focus on Canvas

SCHEDULING EXAMPLE: HIGH-LEVEL VIEW

- 3 Processes in Chrome:
 - **P1**: Download movies
 - **P2**: Open Canvas
 - **P3**: Search StackOverflow

- Example

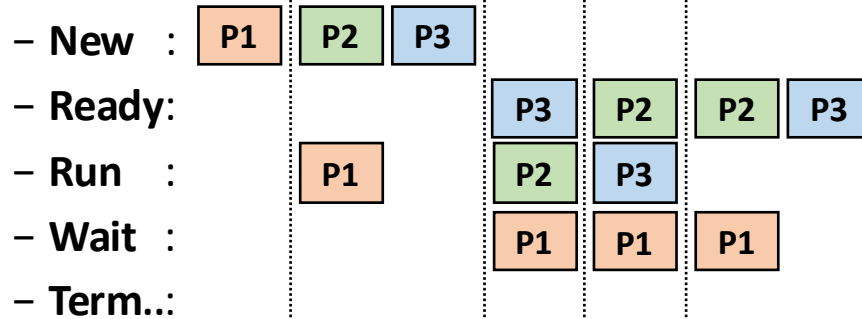


- **Scenario**: while downloading, you start searching StackOverflow

SCHEDULING EXAMPLE: HIGH-LEVEL VIEW

- 3 Processes in Chrome:
 - **P1**: Download movies
 - **P2**: Open Canvas
 - **P3**: Search StackOverflow

- Example

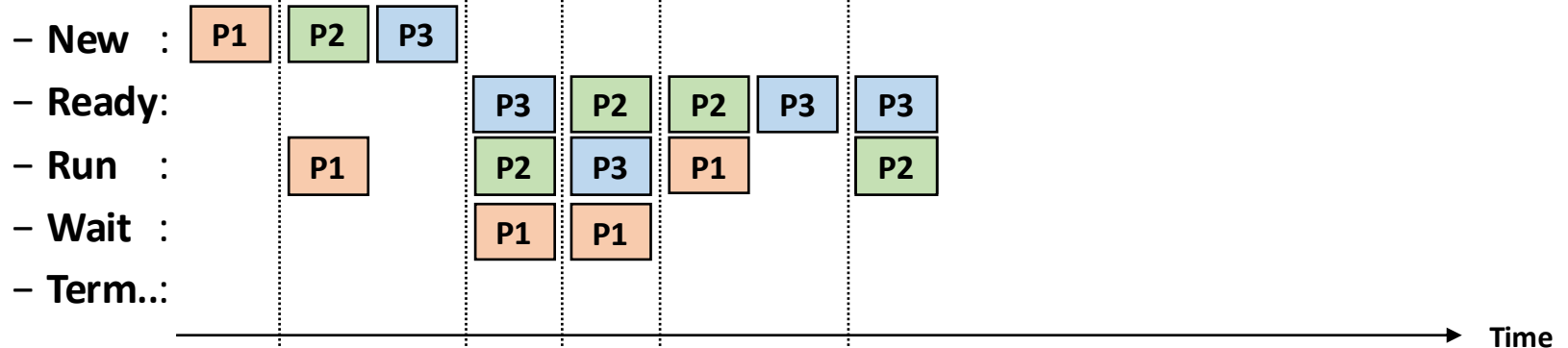


- **Scenario**: downloading movies are done

SCHEDULING EXAMPLE: HIGH-LEVEL VIEW

- 3 Processes in Chrome:
 - **P1**: Download movies
 - **P2**: Open Canvas
 - **P3**: Search StackOverflow

- Example



- **Scenario**: close the download tab, and keep looking at Canvas

TOPICS FOR TODAY

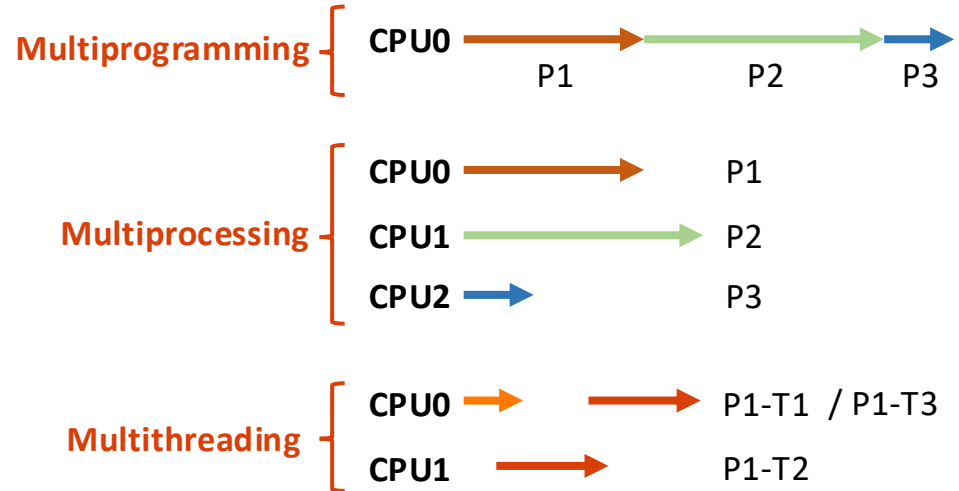
- Part I: Context switching
 - Provide abstraction
 - Why do we need context switching?
 - What is a context?
 - How does OS perform a context switch?
 - Offer standard libraries
 - How do we observe a context switch?
 - How do we trigger a context switch manually?
 - How do we save and restore a context ourselves?
 - Manage resources
 - (Note) We will talk “who gets the GPU next” in the scheduling class
 - (Note) We will talk about “shared resource conflicts” in the synchronization class

PRELIMINARIES ON TERMINOLOGY

- **Definitions:**

- Multiprogramming vs. multi-processing vs. multi-threading

- Multi-programming: multiple jobs (or processes)
- Multi-processing: multiple processors (CPUs)
- Multi-threading: multiple threads



WHAT IS CONTEXT SWITCHING?

- **Context switching**
 - **Definition:** OS stores the current process's status and loads the new process's one
 - **Informal:** OS takes a CPU from one process and gives it to another

WHAT IS A (PROCESS/THREAD) CONTEXT?

- (Linux) Process context in x86 Intel CPU architectures

– Code

- Program counter (PC)
- Instruction pointer (EIP)

– Stack and heap

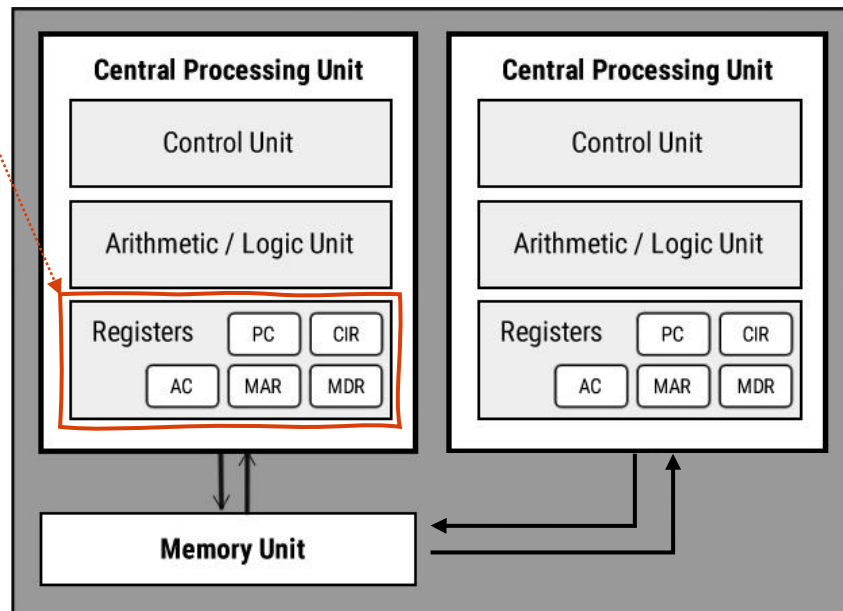
- Stack pointer (ESP, EBP)
- Heap pointer (Page table)

– Running context

- Process state (ID, ...)
- Execution flags
- CPU # to run
- Scheduling policy
- Mem. virtualization

– ...

Process Context: A set of information that OS requires to run a process on a CPU, different from CPU vendors (ex. In Linux, it's defined as *task_struct*, [link](#))



WHAT IS A (PROCESS/THREAD) CONTEXT?

- (Linux) defines this process context

– Code

- Program counter
- Instruction pointer

– Stack and heap

- Stack pointer
- Heap pointer

– Running context

- Process state (ID, ...)
- Execution flags
- CPU # to run
- Scheduling policy
- Mem. virtualization

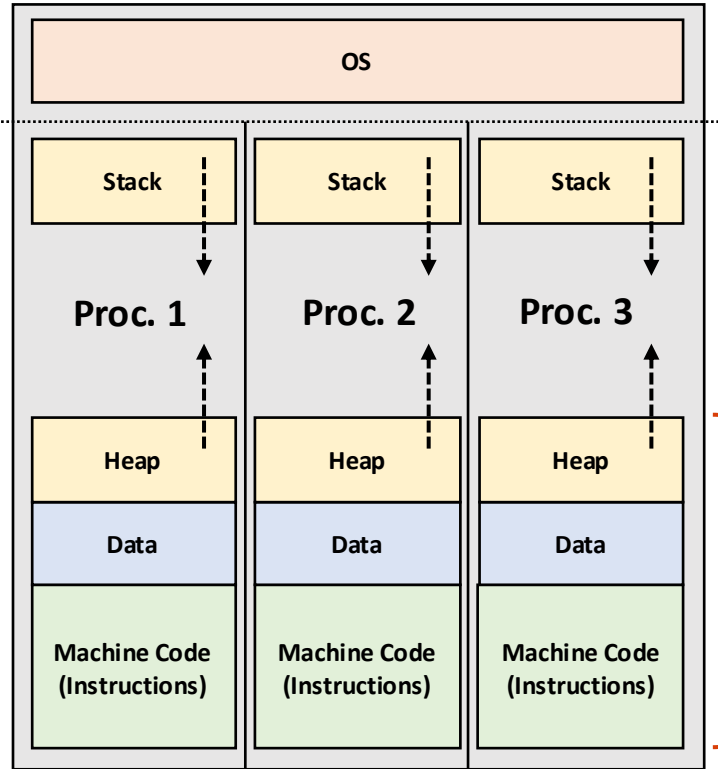
– ...

Process Context: A set of information that OS requires to run a process on a CPU, different from CPU vendors (ex. In Linux, it's defined as *task_struct*, [link](#))

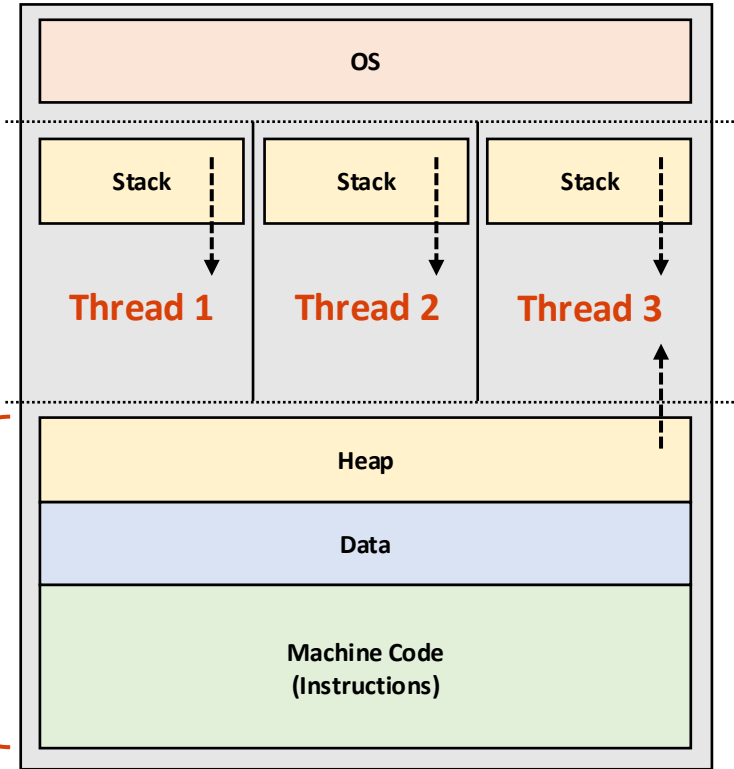
```
728 struct task_struct {
729     #ifdef CONFIG_THREAD_INFO_IN_TASK
730         /*
731          * For reasons of header soup (see current_thread_info()), this
732          * must be the first element of task_struct.
733          */
734         struct thread_info      thread_info;
735     #endif
736     unsigned int                __state;
737
738     #ifdef CONFIG_PREEMPT_RT
739         /* saved state for "spinlock sleepers" */
740         unsigned int            saved_state;
741     #endif
742
743     /*
744      * This begins the randomizable portion of task_struct. Only
745      * scheduling-critical items should be added above here.
746      */
747     randomized_struct_fields_start
748
749     void                        *stack;
750     refcount_t                  usage;
751     /* Per task flags (PF_*), defined further below: */
752     unsigned int                flags;
753     unsigned int                ptrace;
754
755     struct sched_info           sched_info;
756
757     struct list_head           tasks;
758     #ifdef CONFIG_SMP
759     struct plist_node          pushable_tasks;
760     struct rb_node             pushable_dl_tasks;
761     #endif
762
763     struct mm_struct           *mm;
764     struct mm_struct           *active_mm;
765
766     /* Per-thread vma caching: */
767     struct vmacache            vmacache;
768
769     #ifdef SPLIT_RSS_COUNTING
770     struct task_rss_stat       rss_stat;
771     #endif
772
773     int                        exit_state;
774     int                        exit_code;
775     int                        exit_signal;
776     /* The signal sent when the parent dies: */
777     int                        pdeath_signal;
778     /* JOBCTL_*, siglock protected: */
779     unsigned long              jobctl;
780
781     /* Used for emulating ABI behavior of previous Linux versions: */
782     unsigned int               personality;
783 }
```

RECAP: PROCESS VS. THREAD

Processes on memory



Threads in a process on memory



Reduce Duplications

WHAT IS A (PROCESS/THREAD) CONTEXT?

- (Linux) has “thread control block”

– Code

- Program counter
- Instruction pointer

– Stack and heap

- Stack pointer
- Heap pointer

– Running context

- Process state (ID, ...)
- Execution flags
- CPU # to run
- Scheduling policy
- Mem. virtualization

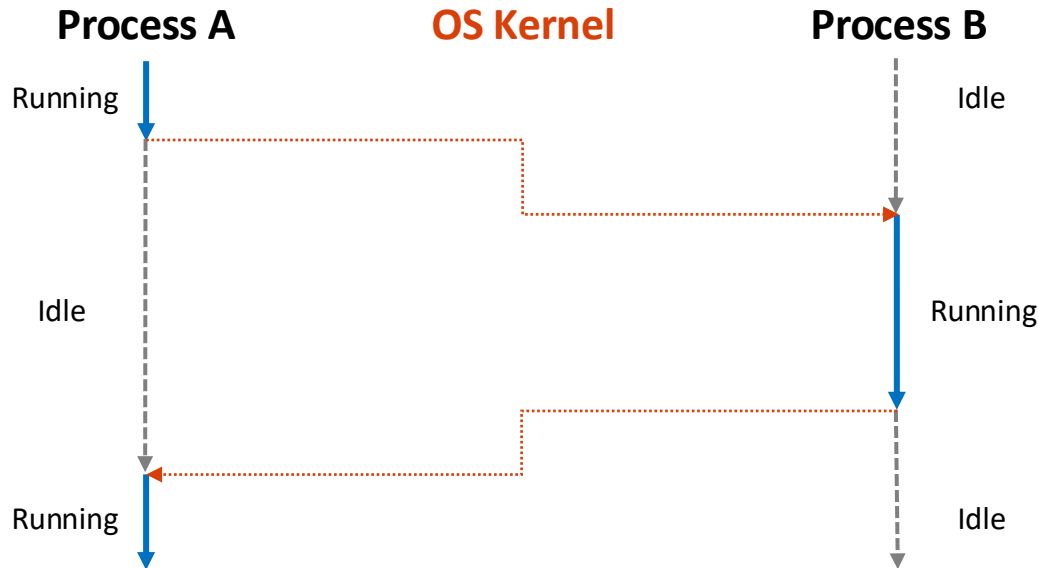
Thread Control Block: A set of information that OS requires to run a thread on a CPU, different from CPU vendors (ex. In Linux, it's the same: *task_struct*, [link](#))

```
728 struct task_struct {
729     #ifdef CONFIG_THREAD_INFO_IN_TASK
730         /*
731          * For reasons of header soup (see current_thread_info()), this
732          * must be the first element of task_struct.
733          */
734         struct thread_info      thread_info;
735     #endif
736     unsigned int                __state;
737
738     #ifdef CONFIG_PREEMPT_RT
739         /* saved state for "spinlock sleepers" */
740         unsigned int            saved_state;
741     #endif
742
743     /*
744      * This begins the randomizable portion of task_struct. Only
745      * scheduling-critical items should be added above here.
746      */
747     randomized_struct_fields_start
748
749     void                        *stack;
750     refcount_t                  usage;
751     /* Per task flags (PF_*), defined further below: */
752     unsigned int                flags;
753     unsigned int                ptrace;
754
755     struct sched_info           sched_info;
756
757     struct list_head            tasks;
758     #ifdef CONFIG_SMP
759     struct plist_node            pushable_tasks;
760     struct rb_node              pushable_dl_tasks;
761     #endif
762
763     struct mm_struct            *mm;
764     struct mm_struct            *active_mm;
765
766     /* Per-thread vma caching: */
767     struct vmacache             vmacache;
768
769     #ifdef SPLIT_RSS_COUNTING
770     struct task_rss_stat         rss_stat;
771     #endif
772
773     int                          exit_state;
774     int                          exit_code;
775     int                          exit_signal;
776     /* The signal sent when the parent dies: */
777     int                          pdeath_signal;
778     /* JOBCTL_*, siglock protected: */
779     unsigned long               jobctl;
780
781     /* Previous Linux versions: */
782     int                          pid;
783     int                          uid;
784     int                          gid;
785     int                          euid;
786     int                          egid;
787     int                          fsuid;
788     int                          fsgid;
789     int                          iuid;
790     int                          igid;
791     int                          suid;
792     int                          sgid;
793     int                          uid_val;
794     int                          gid_val;
795     int                          fsuid_val;
796     int                          fsgid_val;
797     int                          iuid_val;
798     int                          igid_val;
799     int                          suid_val;
800     int                          sgid_val;
801     int                          uid_val0;
802     int                          gid_val0;
803     int                          fsuid_val0;
804     int                          fsgid_val0;
805     int                          iuid_val0;
806     int                          igid_val0;
807     int                          suid_val0;
808     int                          sgid_val0;
809     int                          uid_val1;
810     int                          gid_val1;
811     int                          fsuid_val1;
812     int                          fsgid_val1;
813     int                          iuid_val1;
814     int                          igid_val1;
815     int                          suid_val1;
816     int                          sgid_val1;
817     int                          uid_val2;
818     int                          gid_val2;
819     int                          fsuid_val2;
820     int                          fsgid_val2;
821     int                          iuid_val2;
822     int                          igid_val2;
823     int                          suid_val2;
824     int                          sgid_val2;
825     int                          uid_val3;
826     int                          gid_val3;
827     int                          fsuid_val3;
828     int                          fsgid_val3;
829     int                          iuid_val3;
830     int                          igid_val3;
831     int                          suid_val3;
832     int                          sgid_val3;
833     int                          uid_val4;
834     int                          gid_val4;
835     int                          fsuid_val4;
836     int                          fsgid_val4;
837     int                          iuid_val4;
838     int                          igid_val4;
839     int                          suid_val4;
840     int                          sgid_val4;
841     int                          uid_val5;
842     int                          gid_val5;
843     int                          fsuid_val5;
844     int                          fsgid_val5;
845     int                          iuid_val5;
846     int                          igid_val5;
847     int                          suid_val5;
848     int                          sgid_val5;
849     int                          uid_val6;
850     int                          gid_val6;
851     int                          fsuid_val6;
852     int                          fsgid_val6;
853     int                          iuid_val6;
854     int                          igid_val6;
855     int                          suid_val6;
856     int                          sgid_val6;
857     int                          uid_val7;
858     int                          gid_val7;
859     int                          fsuid_val7;
860     int                          fsgid_val7;
861     int                          iuid_val7;
862     int                          igid_val7;
863     int                          suid_val7;
864     int                          sgid_val7;
865     int                          uid_val8;
866     int                          gid_val8;
867     int                          fsuid_val8;
868     int                          fsgid_val8;
869     int                          iuid_val8;
870     int                          igid_val8;
871     int                          suid_val8;
872     int                          sgid_val8;
873     int                          uid_val9;
874     int                          gid_val9;
875     int                          fsuid_val9;
876     int                          fsgid_val9;
877     int                          iuid_val9;
878     int                          igid_val9;
879     int                          suid_val9;
880     int                          sgid_val9;
881     int                          uid_val10;
882     int                          gid_val10;
883     int                          fsuid_val10;
884     int                          fsgid_val10;
885     int                          iuid_val10;
886     int                          igid_val10;
887     int                          suid_val10;
888     int                          sgid_val10;
889     int                          uid_val11;
890     int                          gid_val11;
891     int                          fsuid_val11;
892     int                          fsgid_val11;
893     int                          iuid_val11;
894     int                          igid_val11;
895     int                          suid_val11;
896     int                          sgid_val11;
897     int                          uid_val12;
898     int                          gid_val12;
899     int                          fsuid_val12;
900     int                          fsgid_val12;
901     int                          iuid_val12;
902     int                          igid_val12;
903     int                          suid_val12;
904     int                          sgid_val12;
905     int                          uid_val13;
906     int                          gid_val13;
907     int                          fsuid_val13;
908     int                          fsgid_val13;
909     int                          iuid_val13;
910     int                          igid_val13;
911     int                          suid_val13;
912     int                          sgid_val13;
913     int                          uid_val14;
914     int                          gid_val14;
915     int                          fsuid_val14;
916     int                          fsgid_val14;
917     int                          iuid_val14;
918     int                          igid_val14;
919     int                          suid_val14;
920     int                          sgid_val14;
921     int                          uid_val15;
922     int                          gid_val15;
923     int                          fsuid_val15;
924     int                          fsgid_val15;
925     int                          iuid_val15;
926     int                          igid_val15;
927     int                          suid_val15;
928     int                          sgid_val15;
929     int                          uid_val16;
930     int                          gid_val16;
931     int                          fsuid_val16;
932     int                          fsgid_val16;
933     int                          iuid_val16;
934     int                          igid_val16;
935     int                          suid_val16;
936     int                          sgid_val16;
937     int                          uid_val17;
938     int                          gid_val17;
939     int                          fsuid_val17;
940     int                          fsgid_val17;
941     int                          iuid_val17;
942     int                          igid_val17;
943     int                          suid_val17;
944     int                          sgid_val17;
945     int                          uid_val18;
946     int                          gid_val18;
947     int                          fsuid_val18;
948     int                          fsgid_val18;
949     int                          iuid_val18;
950     int                          igid_val18;
951     int                          suid_val18;
952     int                          sgid_val18;
953     int                          uid_val19;
954     int                          gid_val19;
955     int                          fsuid_val19;
956     int                          fsgid_val19;
957     int                          iuid_val19;
958     int                          igid_val19;
959     int                          suid_val19;
960     int                          sgid_val19;
961     int                          uid_val20;
962     int                          gid_val20;
963     int                          fsuid_val20;
964     int                          fsgid_val20;
965     int                          iuid_val20;
966     int                          igid_val20;
967     int                          suid_val20;
968     int                          sgid_val20;
969     int                          uid_val21;
970     int                          gid_val21;
971     int                          fsuid_val21;
972     int                          fsgid_val21;
973     int                          iuid_val21;
974     int                          igid_val21;
975     int                          suid_val21;
976     int                          sgid_val21;
977     int                          uid_val22;
978     int                          gid_val22;
979     int                          fsuid_val22;
980     int                          fsgid_val22;
981     int                          iuid_val22;
982     int                          igid_val22;
983     int                          suid_val22;
984     int                          sgid_val22;
985     int                          uid_val23;
986     int                          gid_val23;
987     int                          fsuid_val23;
988     int                          fsgid_val23;
989     int                          iuid_val23;
990     int                          igid_val23;
991     int                          suid_val23;
992     int                          sgid_val23;
993     int                          uid_val24;
994     int                          gid_val24;
995     int                          fsuid_val24;
996     int                          fsgid_val24;
997     int                          iuid_val24;
998     int                          igid_val24;
999     int                          suid_val24;
1000    int                          sgid_val24;
1001    int                          uid_val25;
1002    int                          gid_val25;
1003    int                          fsuid_val25;
1004    int                          fsgid_val25;
1005    int                          iuid_val25;
1006    int                          igid_val25;
1007    int                          suid_val25;
1008    int                          sgid_val25;
1009    int                          uid_val26;
1010    int                          gid_val26;
1011    int                          fsuid_val26;
1012    int                          fsgid_val26;
1013    int                          iuid_val26;
1014    int                          igid_val26;
1015    int                          suid_val26;
1016    int                          sgid_val26;
1017    int                          uid_val27;
1018    int                          gid_val27;
1019    int                          fsuid_val27;
1020    int                          fsgid_val27;
1021    int                          iuid_val27;
1022    int                          igid_val27;
1023    int                          suid_val27;
1024    int                          sgid_val27;
1025    int                          uid_val28;
1026    int                          gid_val28;
1027    int                          fsuid_val28;
1028    int                          fsgid_val28;
1029    int                          iuid_val28;
1030    int                          igid_val28;
1031    int                          suid_val28;
1032    int                          sgid_val28;
1033    int                          uid_val29;
1034    int                          gid_val29;
1035    int                          fsuid_val29;
1036    int                          fsgid_val29;
1037    int                          iuid_val29;
1038    int                          igid_val29;
1039    int                          suid_val29;
1040    int                          sgid_val29;
1041    int                          uid_val30;
1042    int                          gid_val30;
1043    int                          fsuid_val30;
1044    int                          fsgid_val30;
1045    int                          iuid_val30;
1046    int                          igid_val30;
1047    int                          suid_val30;
1048    int                          sgid_val30;
1049    int                          uid_val31;
1050    int                          gid_val31;
1051    int                          fsuid_val31;
1052    int                          fsgid_val31;
1053    int                          iuid_val31;
1054    int                          igid_val31;
1055    int                          suid_val31;
1056    int                          sgid_val31;
1057    int                          uid_val32;
1058    int                          gid_val32;
1059    int                          fsuid_val32;
1060    int                          fsgid_val32;
1061    int                          iuid_val32;
1062    int                          igid_val32;
1063    int                          suid_val32;
1064    int                          sgid_val32;
1065    int                          uid_val33;
1066    int                          gid_val33;
1067    int                          fsuid_val33;
1068    int                          fsgid_val33;
1069    int                          iuid_val33;
1070    int                          igid_val33;
1071    int                          suid_val33;
1072    int                          sgid_val33;
1073    int                          uid_val34;
1074    int                          gid_val34;
1075    int                          fsuid_val34;
1076    int                          fsgid_val34;
1077    int                          iuid_val34;
1078    int                          igid_val34;
1079    int                          suid_val34;
1080    int                          sgid_val34;
1081    int                          uid_val35;
1082    int                          gid_val35;
1083    int                          fsuid_val35;
1084    int                          fsgid_val35;
1085    int                          iuid_val35;
1086    int                          igid_val35;
1087    int                          suid_val35;
1088    int                          sgid_val35;
1089    int                          uid_val36;
1090    int                          gid_val36;
1091    int                          fsuid_val36;
1092    int                          fsgid_val36;
1093    int                          iuid_val36;
1094    int                          igid_val36;
1095    int                          suid_val36;
1096    int                          sgid_val36;
1097    int                          uid_val37;
1098    int                          gid_val37;
1099    int                          fsuid_val37;
1100    int                          fsgid_val37;
1101    int                          iuid_val37;
1102    int                          igid_val37;
1103    int                          suid_val37;
1104    int                          sgid_val37;
1105    int                          uid_val38;
1106    int                          gid_val38;
1107    int                          fsuid_val38;
1108    int                          fsgid_val38;
1109    int                          iuid_val38;
1110    int                          igid_val38;
1111    int                          suid_val38;
1112    int                          sgid_val38;
1113    int                          uid_val39;
1114    int                          gid_val39;
1115    int                          fsuid_val39;
1116    int                          fsgid_val39;
1117    int                          iuid_val39;
1118    int                          igid_val39;
1119    int                          suid_val39;
1120    int                          sgid_val39;
1121    int                          uid_val40;
1122    int                          gid_val40;
1123    int                          fsuid_val40;
1124    int                          fsgid_val40;
1125    int                          iuid_val40;
1126    int                          igid_val40;
1127    int                          suid_val40;
1128    int                          sgid_val40;
1129    int                          uid_val41;
1130    int                          gid_val41;
1131    int                          fsuid_val41;
1132    int                          fsgid_val41;
1133    int                          iuid_val41;
1134    int                          igid_val41;
1135    int                          suid_val41;
1136    int                          sgid_val41;
1137    int                          uid_val42;
1138    int                          gid_val42;
1139    int                          fsuid_val42;
1140    int                          fsgid_val42;
1141    int                          iuid_val42;
1142    int                          igid_val42;
1143    int                          suid_val42;
1144    int                          sgid_val42;
1145    int                          uid_val43;
1146    int                          gid_val43;
1147    int                          fsuid_val43;
1148    int                          fsgid_val43;
1149    int                          iuid_val43;
1150    int                          igid_val43;
1151    int                          suid_val43;
1152    int                          sgid_val43;
1153    int                          uid_val44;
1154    int                          gid_val44;
1155    int                          fsuid_val44;
1156    int                          fsgid_val44;
1157    int                          iuid_val44;
1158    int                          igid_val44;
1159    int                          suid_val44;
1160    int                          sgid_val44;
1161    int                          uid_val45;
1162    int                          gid_val45;
1163    int                          fsuid_val45;
1164    int                          fsgid_val45;
1165    int                          iuid_val45;
1166    int                          igid_val45;
1167    int                          suid_val45;
1168    int                          sgid_val45;
1169    int                          uid_val46;
1170    int                          gid_val46;
1171    int                          fsuid_val46;
1172    int                          fsgid_val46;
1173    int                          iuid_val46;
1174    int                          igid_val46;
1175    int                          suid_val46;
1176    int                          sgid_val46;
1177    int                          uid_val47;
1178    int                          gid_val47;
1179    int                          fsuid_val47;
1180    int                          fsgid_val47;
1181    int                          iuid_val47;
1182    int                          igid_val47;
1183    int                          suid_val47;
1184    int                          sgid_val47;
1185    int                          uid_val48;
1186    int                          gid_val48;
1187    int                          fsuid_val48;
1188    int                          fsgid_val48;
1189    int                          iuid_val48;
1190    int                          igid_val48;
1191    int                          suid_val48;
1192    int                          sgid_val48;
1193    int                          uid_val49;
1194    int                          gid_val49;
1195    int                          fsuid_val49;
1196    int                          fsgid_val49;
1197    int                          iuid_val49;
1198    int                          igid_val49;
1199    int                          suid_val49;
1200    int                          sgid_val49;
1201    int                          uid_val50;
1202    int                          gid_val50;
1203    int                          fsuid_val50;
1204    int                          fsgid_val50;
1205    int                          iuid_val50;
1206    int                          igid_val50;
1207    int                          suid_val50;
1208    int                          sgid_val50;
1209    int                          uid_val51;
1210    int                          gid_val51;
1211    int                          fsuid_val51;
1212    int                          fsgid_val51;
1213    int                          iuid_val51;
1214    int                          igid_val51;
1215    int                          suid_val51;
1216    int                          sgid_val51;
1217    int                          uid_val52;
1218    int                          gid_val52;
1219    int                          fsuid_val52;
1220    int                          fsgid_val52;
1221    int                          iuid_val52;
1222    int                          igid_val52;
1223    int                          suid_val52;
1224    int                          sgid_val52;
1225    int                          uid_val53;
1226    int                          gid_val53;
1227    int                          fsuid_val53;
1228    int                          fsgid_val53;
1229    int                          iuid_val53;
1230    int                          igid_val53;
1231    int                          suid_val53;
1232    int                          sgid_val53;
1233    int                          uid_val54;
1234    int                          gid_val54;
1235    int                          fsuid_val54;
1236    int                          fsgid_val54;
1237    int                          iuid_val54;
1238    int                          igid_val54;
1239    int                          suid_val54;
1240    int                          sgid_val54;
1241    int                          uid_val55;
1242    int                          gid_val55;
1243    int                          fsuid_val55;
1244    int                          fsgid_val55;
1245    int                          iuid_val55;
1246    int                          igid_val55;
1247    int                          suid_val55;
1248    int                          sgid_val55;
1249    int                          uid_val56;
1250    int                          gid_val56;
1251    int                          fsuid_val56;
1252    int                          fsgid_val56;
1253    int                          iuid_val56;
1254    int                          igid_val56;
1255    int                          suid_val56;
1256    int                          sgid_val56;
1257    int                          uid_val57;
1258    int                          gid_val57;
1259    int                          fsuid_val57;
1260    int                          fsgid_val57;
1261    int                          iuid_val57;
1262    int                          igid_val57;
1263    int                          suid_val57;
1264    int                          sgid_val57;
1265    int                          uid_val58;
1266    int                          gid_val58;
1267    int                          fsuid_val58;
1268    int                          fsgid_val58;
1269    int                          iuid_val58;
1270    int                          igid_val58;
1271    int                          suid_val58;
1272    int                          sgid_val58;
1273    int                          uid_val59;
1274    int                          gid_val59;
1275    int                          fsuid_val59;
1276    int                          fsgid_val59;
1277    int                          iuid_val59;
1278    int                          igid_val59;
1279    int                          suid_val59;
1280    int                          sgid_val59;
1281    int                          uid_val60;
1282    int                          gid_val60;
1283    int                          fsuid_val60;
1284    int                          fsgid_val60;
1285    int                          iuid_val60;
1286    int                          igid_val60;
1287    int                          suid_val60;
1288    int                          sgid_val60;
1289    int                          uid_val61;
1290    int                          gid_val61;
1291    int                          fsuid_val61;
1292    int                          fsgid_val61;
1293    int                          iuid_val61;
1294    int                          igid_val61;
1295    int                          suid_val61;
1296    int                          sgid_val61;
1297    int                          uid_val62;
1298    int                          gid_val62;
1299    int                          fsuid_val62;
1300    int                          fsgid_val62;
1301    int                          iuid_val62;
1302    int                          igid_val62;
1303    int                          suid_val62;
1304    int                          sgid_val62;
1305    int                          uid_val63;
1306    int                          gid_val63;
1307    int                          fsuid_val63;
1308    int                          fsgid_val63;
1309    int                          iuid_val63;
1310    int                          igid_val63;
1311    int                          suid_val63;
1312    int                          sgid_val63;
1313    int                          uid_val64;
1314    int                          gid_val64;
1315    int                          fsuid_val64;
1316    int                          fsgid_val64;
1317    int                          iuid_val64;
1318    int                          igid_val64;
1319    int                          suid_val64;
1320    int                          sgid_val64;
1321    int                          uid_val65;
1322    int                          gid_val65;
1323    int                          fsuid_val65;
1324    int                          fsgid_val65;
1325    int                          iuid_val65;
1326    int                          igid_val65;
1327    int                          suid_val65;
1328    int                          sgid_val65;
1329    int                          uid_val66;
1330    int                          gid_val66;
1331    int                          fsuid_val66;
1332    int                          fsgid_val66;
1333    int                          iuid_val66;
1334    int                          igid_val66;
1335    int                          suid_val66;
1336    int                          sgid_val66;
1337    int                          uid_val67;
1338    int                          gid_val67;
1339    int                          fsuid_val67;
1340    int                          fsgid_val67;
1341    int                          iuid_val67;
1342    int                          igid_val67;
1343    int                          suid_val67;
1344    int                          sgid_val67;
1345    int                          uid_val68;
1346    int                          gid_val68;
1347    int                          fsuid_val68;
1348    int                          fsgid_val68;
1349    int                          iuid_val68;
1350    int                          igid_val68;
1351    int                          suid_val68;
1352    int                          sgid_val68;
1353    int                          uid_val69;
1354    int                          gid_val69;
1355    int                          fsuid_val69;
1356    int                          fsgid_val69;
1357    int                          iuid_val69;
1358    int                          igid_val69;
1359    int                          suid_val69;
1360    int                          sgid_val69;
1361    int                          uid_val70;
1362    int                          gid_val70;
1363    int                          fsuid_val70;
1364    int                          fsgid_val70;
1365    int                          iuid_val70;
1366    int                          igid_val70;
1367    int                          suid_val70;
1368    int                          sgid_val70;
1369    int                          uid_val71;
1370    int                          gid_val71;
1371    int                          fsuid_val71;
1372    int                          fsgid_val71;
1373    int                          iuid_val71;
1374    int                          igid_val71;
1375    int                          suid_val71;
1376    int                          sgid_val71;
1377    int                          uid_val72;
1378    int                          gid_val72;
1379    int                          fsuid_val72;
1380    int                          fsgid_val72;
1381    int                          iuid_val72;
1382    int                          igid_val72;
1383    int                          suid_val72;
1384    int                          sgid_val72;
1385    int                          uid_val73;
1386    int                          gid_val73;
1387    int                          fsuid_val73;
1388    int                          fsgid_val73;
1389    int                          iuid_val73;
1390    int                          igid_val73;
1391    int                          suid_val73;
1392    int                          sgid_val73;
1393    int                          uid_val74;
1394    int                          gid_val74;
1395    int                          fsuid_val74;
1396    int                          fsgid_val74;
1397    int                          iuid_val74;
1398    int                          igid_val74;
1399    int                          suid_val74;
1400    int                          sgid_val74;
1401    int                          uid_val75;
1402    int                          gid_val75;
1403    int                          fsuid_val75;
1404    int                          fsgid_val75;
1405    int                          iuid_val75;
1406    int                          igid_val75;
1407    int                          suid_val75;
1408    int                          sgid_val75;
1409    int                          uid_val76;
1410    int                          gid_val76;
1411    int                          fsuid_val76;
1412    int                          fsgid_val76;
1413    int                          iuid_val76;
1414    int                          igid_val76;
1415    int                          suid_val76;
1416    int                          sgid_val76;
1417    int                          uid_val77;
1418    int                          gid_val77;
1419    int                          fsuid_val77;
1420    int                          fsgid_val77;
1421    int                          iuid_val77;
1422    int                          igid_val77;
1423    int                          suid_val77;
1424    int                          sgid_val77;
1425    int                          uid_val78;
1426    int                          gid_val78;
1427    int                          fsuid_val78;
1428    int                          fsgid_val78;
1429    int                          iuid_val78;
1430    int                          igid_val78;
1431    int                          suid_val78;
1432    int                          sgid_val78;
1433    int                          uid_val79;
1434    int                          gid_val79;
1435    int                          fsuid_val79;
1436    int                          fsgid_val79;
1437    int                          iuid_val79;
1438    int                          igid_val79;
1439    int                          suid_val79;
1440    int                          sgid_val79;
1441    int                          uid_val80;
1442    int                          gid_val80;
1443    int                          fsuid_val80;
1444    int                          fsgid_val80;
1445    int                          iuid_val80;
1446    int                          igid_val80;
1447    int                          suid_val80;
1448    int                          sgid_val80;
1449    int                          uid_val81;
1450    int                          gid_val81;
1451    int                          fsuid_val81;
1452    int                          fsgid_val81;
1453    int                          iuid_val81;
1454    int                          igid_val81;
1455    int                          suid_val81;
1456    int                          sgid_val81;
1457    int                          uid_val82;
1458    int                          gid_val82;
1459    int                          fsuid_val82;
1460    int                          fsgid_val82;
1461    int                          iuid_val82;
1462    int                          igid_val82;
1463    int                          suid_val82;
1464    int                          sgid_val82;
1465    int                          uid_val83;
1466    int                          gid_val83;
1467    int                          fsuid_val83;
1468    int                          fsgid_val83;
1469    int                          iuid_val83;
1470    int                          igid_val83;
1471    int                          suid_val83;
1472    int                          sgid_val83;
1473    int                          uid_val84;
1474    int                          gid_val84;
1475    int                          fsuid_val84;
1476    int                          fsgid_val84;
1477    int                          iuid_val84;
1478    int                          igid_val84;
1479    int                          suid_val84;
1480    int                          sgid_val84;
1481    int                          uid_val85;
1482    int                          gid_val85;
1483    int                          fsuid_val85;
1484    int                          fsgid_val85;
1485    int                          iuid_val85;
1486    int                          igid_val85;
1487    int                          suid_val85;
1488    int                          sgid_val85;
1489    int                          uid_val86;
1490    int                          gid_val86;
1491    int                          fsuid_val86;
1492    int                          fsgid_val86;
1493    int                          iuid_val86;
1494    int                          igid_val86;
1495    int                          suid_val86;
1496    int                          sgid_val86;
1497    int                          uid_val87;
1498    int                          gid_val87;
1499    int                          fsuid_val87;
1500    int                          fsgid_val87;
1501    int                          iuid_val87;
1502    int                          igid_val87;
1503    int                          suid_val87;
1504    int                          sgid_val87;
1505    int                          uid_val88;
1506    int                          gid_val88;
1507    int                          fsuid_val88;
1508    int                          fsgid_val88;
1509    int                          iuid_val88;
1510    int                          igid_val88;
1511    int                          suid_val88;
1512    int                          sgid_val88;
1513    int                          uid_val89;
1514    int                          gid_val89;
1515    int                          fsuid_val89;
1516    int                          fsgid_val89;
1517    int                          iuid_val89;
1518    int                          igid_val89;
1519    int                          suid_val89;
1520    int                          sgid_val89;
1521    int                          uid_val90;
1522    int                          gid_val90;
1523    int                          fsuid_val90;
1524    int                          fsgid_val90;
1525    int                          iuid_val90;
1526    int                          igid_val90;
1527    int                          suid_val90;
1528    int                          sgid_val90;
1529    int                          uid_val91;
1530    int                          gid_val91;
1531    int                          fsuid_val91;
1532    int                          fsgid_val91;
1533    int                          iuid_val91;
1534    int                          igid_val91;
1535    int                          suid_val91;
1536    int                          sgid_val91;
1537    int                          uid_val92;
1538    int                          gid_val92;
1539    int                          fsuid_val92;
1540    int                          fsgid_val92;
1541    int                          iuid_val92;
1542    int                          igid_val92;
1543    int                          suid_val92;
1544    int                          sgid_val92;
1545    int                          uid_val93;
1546    int                          gid_val93;
1547    int                          fsuid_val93;
1548    int                          fsgid_val93;
1549    int                          iuid_val93;
1550    int                          igid_val93;
1551    int                          suid_val93;
1552    int                          sgid_val93;
1553    int                          uid_val94;
1554    int                          gid_val94;
1555    int                          fsuid_val94;
1556    int                          fsgid_val94;
1557    int                          iuid_val94;
1558    int                          igid_val94;
1559    int                          suid_val94;
1560    int                          sgid_val94;
1561    int                          uid_val95;
1562    int                          gid_val95;
1563    int                          fsuid_val95;
1564    int                          fsgid_val95;
1565    int                          iuid_val95;
1566    int                          igid_val95;
1567    int                          suid_val95;
1568    int                          sgid_val95;
1569    int                          uid_val96;
1570    int                          gid_val96;
1571    int                          fsuid_val96;
1572    int                          fsgid_val96;
1573    int                          iuid_val96;
1574    int                          igid_val96;
1575    int                          suid_val96;
1576    int                          sgid_val96;
1577    int                          uid_val97;
1578    int                          gid_val97;
1579    int                          fsuid_val97;
1580    int                          fsgid_val97;
1581    int                          iuid_val97;
1582    int                          igid_val97;
1583    int                          suid_val97;
1584    int                          sgid_val97;
1585    int                          uid_val98;
1586    int                          gid_val98;
1587    int                          fsuid_val98;
1588    int                          fsgid_val98;
1589    int                          iuid_val98;
1590    int                          igid_val98;
1591    int                          suid_val98;
1592    int                          sgid_val98;
1593    int                          uid_val99;
1594    int                          gid_val99;
1595    int                          fsuid_val99;
1596    int                          fsgid_val99;
1597    int                          iuid_val99;
1598    int                          igid_val99;
1599    int                          suid_val99;
1600    int                          sgid_val99;
1601    int                          uid_val100;
1602    int                          gid_val100;
1603    int                          fsuid_val100;
1604    int                          fsgid_val100;
1605    int                          iuid_val100;
1606    int                          igid_val100;
1607    int                          suid_val100;
1608    int                          sgid_val100;
1609    int                          uid_val101;
1610    int                          gid_val101;
1611    int                          fsuid_val101;
1612    int                          fsgid_val101;
1613    int                          iuid_val101;
1614    int                          igid_val101;
1615    int                          suid_val101;
1616    int                          sgid_val101;
1617    int                          uid_val102;
1618    int                          gid_val102;
1619    int                          fsuid_val102;
1620    int                          fsgid_val102;
1621    int                          iuid_val102;
1622
```

HOW DOES OS PERFORM CONTEXT SWITCHING?

- **Context switch**

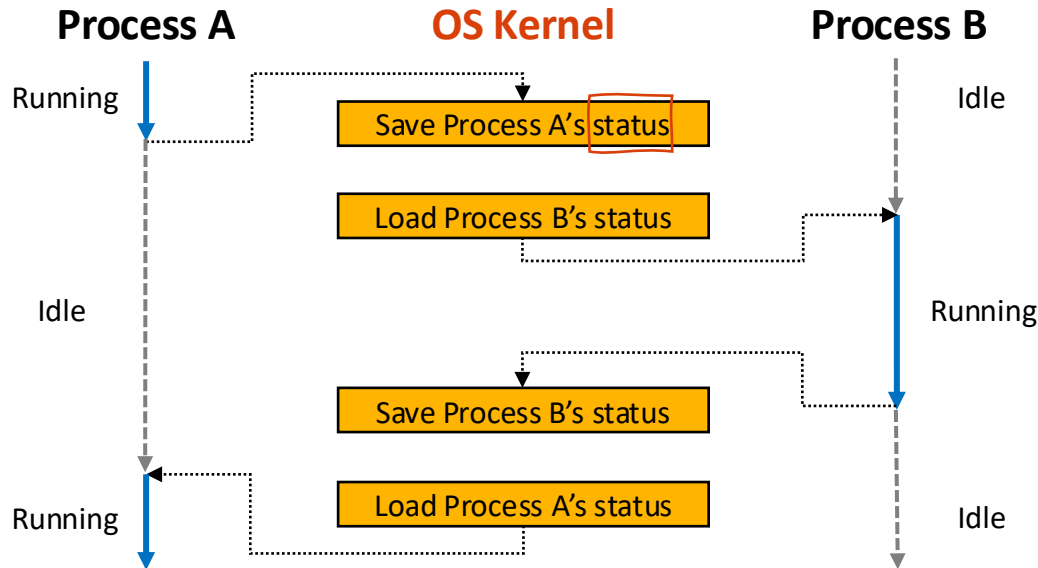
- **Definition:** OS stores the current process's status and loads the new process's one
- **Informal:** OS takes a CPU from one process and gives it to another



HOW DOES OS PERFORM CONTEXT SWITCHING?

- **Context switch**

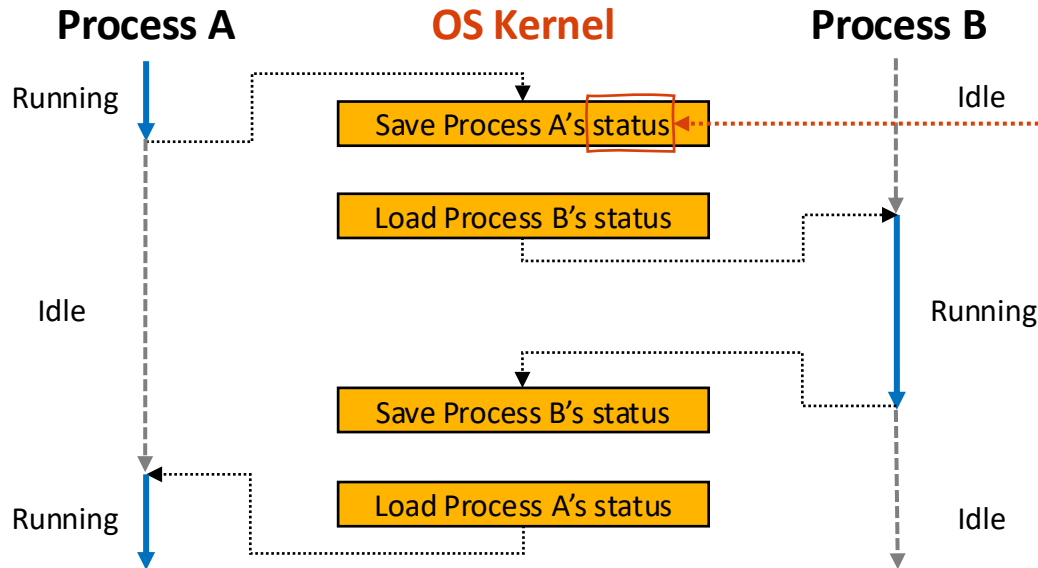
- **Definition:** OS stores the current process's status and loads the new process's one
- **Informal:** OS takes a CPU from one process and gives it to another



HOW DOES OS PERFORM CONTEXT SWITCHING?

- **Context switch**

- **Definition:** OS stores the current process's status and loads the new process's one
- **Informal:** OS takes a CPU from one process and gives it to another



Recall: Process control block

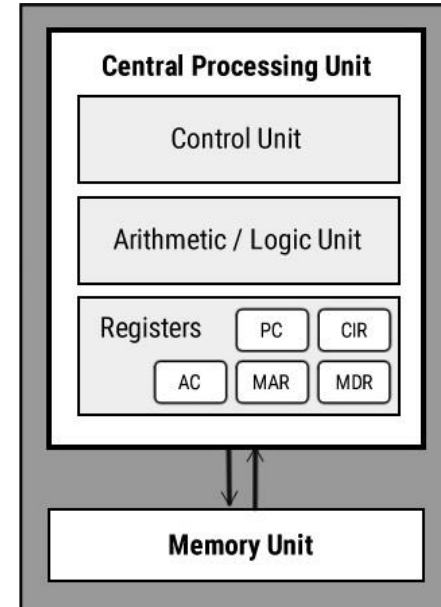
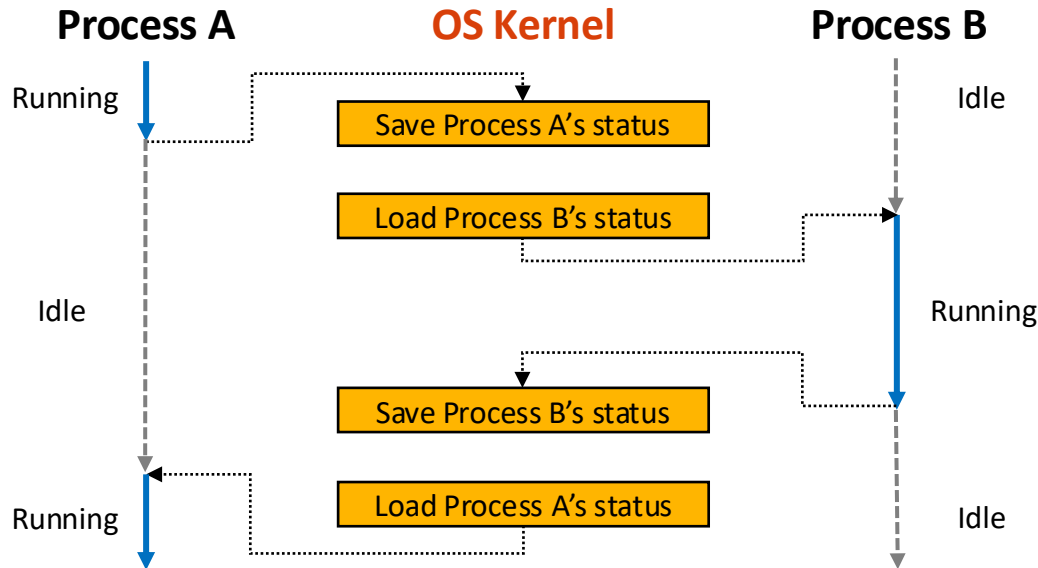
A structure in OS that contains a set of information required to run a process on a CPU. Recall that Linux has *task_struct*.

- CPU#
- Program counter
- Instruction pointer
- Heap/stack pointer
- Process state [!]
- ...

HOW DOES OS PERFORM CONTEXT SWITCHING?

- **Context switch**

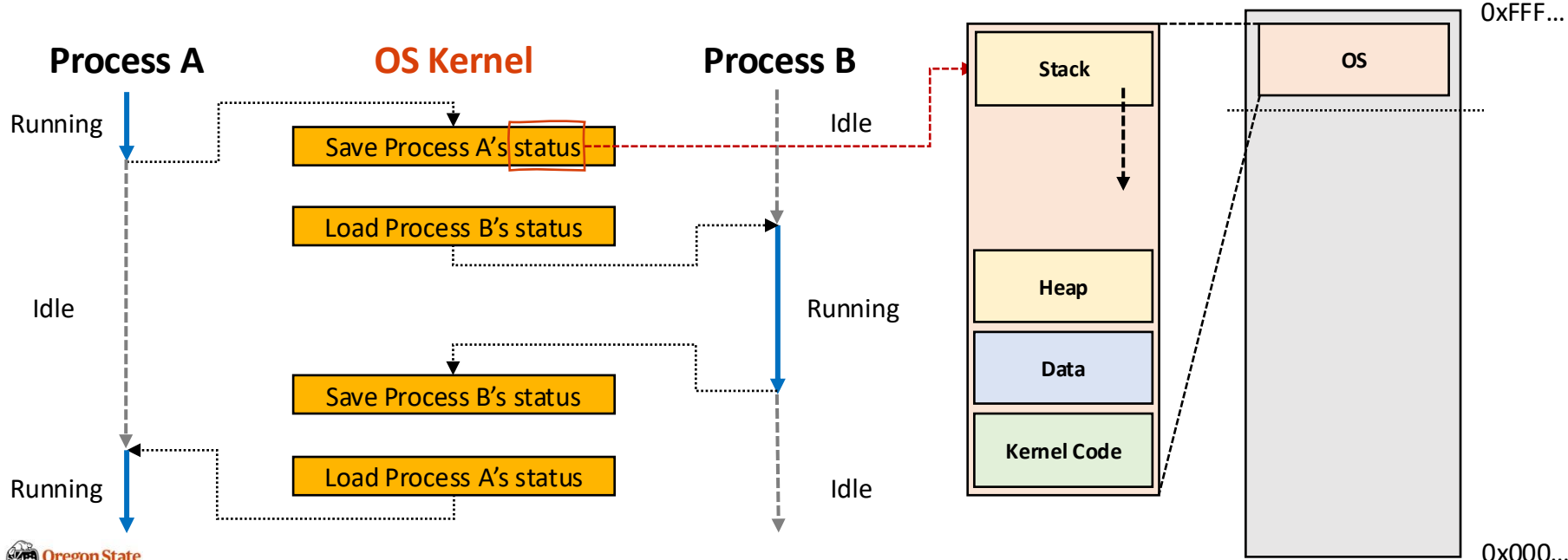
- **Definition:** OS stores the current process's status and loads the new process's one
- **Informal:** OS takes a CPU from one process and gives it to another



WHERE DOES OS STORE THE CONTEXT?

- OS kernel

- **Definition:** a program that has a complete control over the system
- OS kernel stores the context in its own *stack* (*not in the user stack*)



HOW DOES OS PERFORM CONTEXT SWITCHING?

- **Context switch**

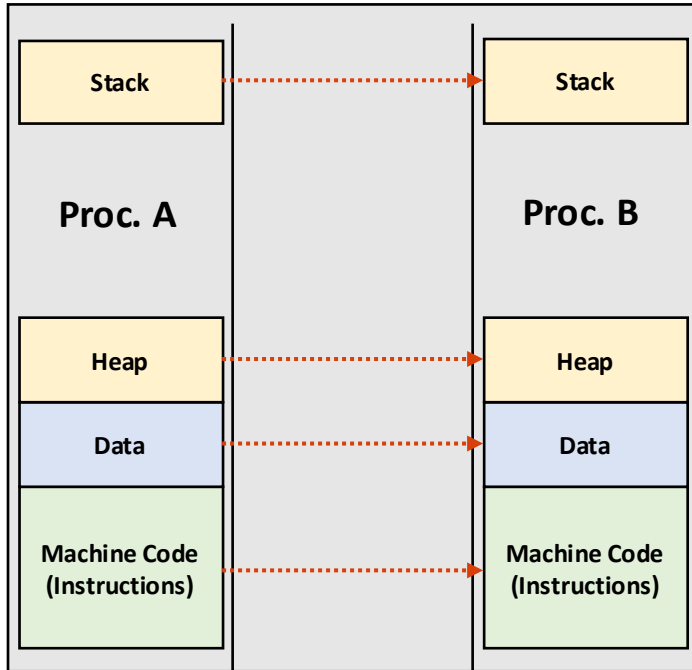
- **Definition:** OS stores the current process's status and loads the new process's one
- **Informal:** OS takes a CPU from one process and gives it to another

- **No free lunch**

- Context switching takes $\sim 5 \mu s$ on average
- OS typically runs 100+ processes
- Too many context switching makes a system unable to respond...

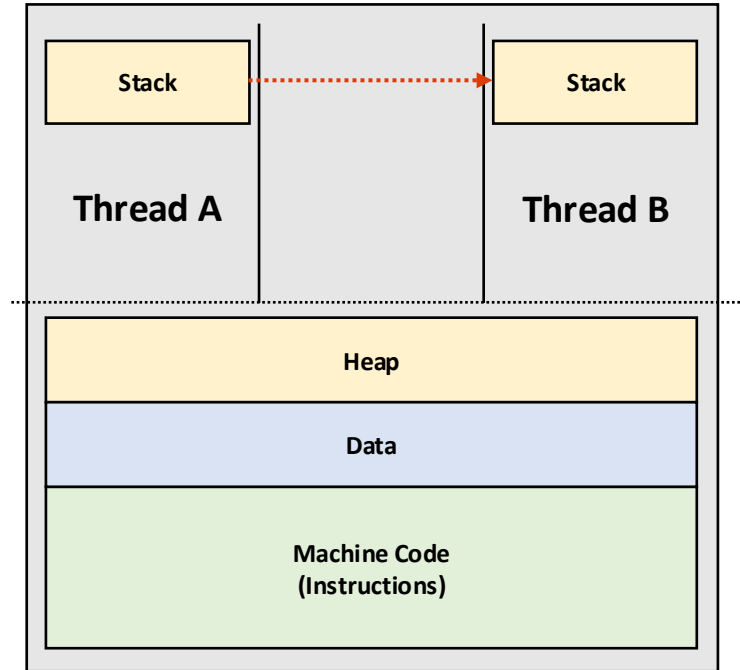
PROCESS VS. THREAD CONTEXT SWITCHING

Process switch



Process switch needs to replace *all*
(4 segments, registers... – needs TLB flush)

Thread switch



Thread switch needs to replace *only*
registers and stack – lightweight

TOPICS FOR TODAY

- Part I: Context switching
 - Provide abstraction
 - Why do we need context switching?
 - What is a context?
 - How does OS perform a context switch?
 - Offer standard libraries
 - How do we observe a context switch?
 - How do we trigger a context switch manually?
 - How do we save and restore a context ourselves?
 - Manage resources
 - (Note) We will talk “who gets the GPU next” in the scheduling class
 - (Note) We will talk about “shared resource conflicts” in the synchronization class

HOW DO WE OBSERVE A CONTEXT SWITCH?

- Process exposes its ctx switch count
 - `/proc/[pid]/status`
 - Example: `$ cat /proc/$$/status | grep ctxt (check twice for difference)`

- `$$` = current (bash) shell process (I/O bound)
 - `voluntary_ctxt_switches` A
 - `nonvoluntary_ctxt_switches` B

- `yes > /dev/null` (Compute bound)
 - `voluntary_ctxt_switches` C
 - `nonvoluntary_ctxt_switches` D

HOW DO WE OBSERVE A CONTEXT SWITCH? – CONT'D

- Trace system calls
 - `strace -c program` or `strace -c -p [pid]`
 - Example: `$ strace -c cat /proc/$$/status`
- Compare these
 - `$$` = current (bash) shell process (I/O bound)
 - `yes > /dev/null` (Compute bound)

HOW DO WE TRIGGER A CONTEXT SWITCH?

- Process can voluntarily give up the CPU
 - `sched_yield()` // in `#include <sched.h>`
- Kernel procedure when we run this example
 - Put the current thread to the thread *queue*
 - OS kernel selects the next thread
 - Note: if there is only a thread in the queue then we can observe only that thread running

```
void *thread_no_yield(void *arg) {
    int id = *(int *)arg;
    for (int i = 0; i < 5; i++) {
        printf("Thread %d: step %d\n", id, i);
    }
    return NULL;
}
```

```
void *thread_with_yield(void *arg) {
    int id = *(int *)arg;
    for (int i = 0; i < 5; i++) {
        printf("Thread %d: step %d\n", id, i);
        sched_yield();
    }
    return NULL;
}
```

```
int main() {
    pthread_t t1, t2;
    int id1 = 1, id2 = 2;
    printf("=== Without sched_yield ===\n");
    pthread_create(&t1, NULL, thread_no_yield, &id1);
    pthread_create(&t2, NULL, thread_no_yield, &id2);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);

    printf("\n=== With sched_yield ===\n");
    pthread_create(&t1, NULL, thread_with_yield, &id1);
    pthread_create(&t2, NULL, thread_with_yield, &id2);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
}
```

HOW DO WE TRIGGER A CONTEXT SWITCH? – CONT'D

- Process can voluntarily give up the CPU
 - `sched_yield()` // in `#include <sched.h>`
 - It increases “voluntary_ctxt_switches”

```
#include <stdio.h>
#include <sched.h>
#include <unistd.h>

int main() {
    pid_t pid = getpid();
    char cmd[100];

    printf("=== Before sched_yield ===\n");
    snprintf(cmd, sizeof(cmd),
             "cat /proc/%d/status | grep ctxt", pid);
    system(cmd);

    for (int i = 0; i < 100; i++) {
        sched_yield();
    }

    printf("\n=== After 100 x sched_yield ===\n");
    system(cmd);

    return 0;
}
```

HOW DO WE SAVE AND RESTORE A CONTEXT?

- POSIX supports
 - `ucontext_t`
 - A structure that stores *only the user stack*
- System calls
 - `getcontext(ucp)`
 - `setcontext(ucp)`
 - `makecontext(ucp, fn, ...)`
 - `swapcontext(oucp, ucp)`

```
typedef struct ucontext_t {  
    struct ucontext_t *uc_link;  
    stack_t          uc_stack;  
    mcontext_t       uc_mcontext;  
    sigset_t         uc_sigmask;  
} ucontext_t;
```

```
ucontext_t ctx_main, ctx_worker;  
char stack[65536];
```

```
void worker() {  
    printf("Worker: step 1\n");  
    swapcontext(&ctx_worker, &ctx_main);  
    printf("Worker: step 2\n");  
}
```

```
int main() {  
    getcontext(&ctx_worker);  
    ctx_worker.uc_stack.ss_sp = stack;  
    ctx_worker.uc_stack.ss_size = sizeof(stack);  
    ctx_worker.uc_link = &ctx_main;  
    makecontext(&ctx_worker, worker, 0);
```

```
    printf("Main: before swap\n");  
    swapcontext(&ctx_main, &ctx_worker);  
    printf("Main: after first swap\n");  
    swapcontext(&ctx_main, &ctx_worker);  
    printf("Main: done\n");
```

```
}
```

TOPICS FOR TODAY

- Part I: Context switching
 - Provide abstraction
 - Why do we need context switching?
 - What is a context?
 - How does OS perform a context switch?
 - Offer standard libraries
 - How do we observe a context switch?
 - How do we trigger a context switch manually?
 - How do we save and restore a context ourselves?
 - Manage resources
 - (Note) We will talk “who gets the GPU next” in the scheduling class
 - (Note) We will talk about “shared resource conflicts” in the synchronization class

Thank You!

M/W 12:00 – 1:50 PM (LINC #200)

Sanghyun Hong

sanghyun.hong@oregonstate.edu



Oregon State
University



TRUE AI
Trustworthy and Responsible AI